ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

11

Criptografía. La ocultación de mensajes y el computador

Vicente Martínez Orga

SINCLAIR AMSTRAD COMMODORE IBM



EDICIONES SIGLO CULTURAL

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

11

Criptografía. La ocultación de mensajes y el ordenador

EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S.A.

Director-editor:
RICARDO ESPAÑOL CRESPO.

Gerente:
ANTONIO G. CUERPO.

Directora de producción:
MARIA LUISA SUAREZ PEREZ.

Directores de la colección:
MANUEL ALFONSECA, Doctor Ingeniero de Telecomunicación y Licenciado en Informática
JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño y maquetación:
BRAVO-LOFISH.

Dibujos:
JOSE OCHOA Y ANTONIO PERERA.

Tomo XI. **Criptografía: La ocultación de mensajes y el ordenador.**
VICENTE MARTINEZ, Doctor en Informática.

Ediciones Siglo Cultural, S.A.

Dirección, redacción y administración:
Sor Angela de la Cruz, 24-7.º G. Teléf. 279 40 36. 28020 Madrid.

Publicidad:
Gofar Publicidad, S.A. Benito de Castro, 12 bis. 28020 Madrid.

Distribución en España:
COEDIS, S.A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.
Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:
CADE, S.R.L. Pasaje Sud América. 1532. Teléf.: 21 24 64.
Buenos Aires - 1.290. Argentina.

ISBN del tomo: 84-7688-035-9.
ISBN de la obra: 84-7688-018-9.

Fotocomposición:
ARTECOMP, S.A. Albarracín, 50. 28037 Madrid.

Imprime:
MATEU CROMO. Pinto (Madrid).

Depósito legal: M-39.892-1986.
Printed in Spain - Impreso en España.

Suscripciones y números atrasados:
Ediciones Siglo Cultural, S.A.
Sor Angela de la Cruz, 24-7.° G. Teléf. 279 40 36. 28020 Madrid

Octubre, 1986.
P.V.P. Canarias: 365,-

INDICE

Los programas que aparecen en este libro funcionan en los ordenadores:

IBM–PC, XT, AT y compatibles.
AMSTRAD–464, 664, 6128, 1512.
SINCLAIR–SPECTRUM 48 K, 128 K, PLUS, PLUS 2.
MSX–Todos los modelos.
COMMODORE–CBM 64 y CBM 128.

PROLOGO

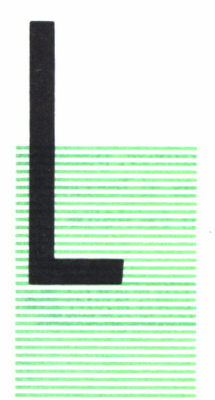
En este libro se ha tratado de recoger, a un nivel comprensible para no avanzados en la materia, distintos sistemas de seguridad, desde los más remotos que se mencionan en la introducción, hasta las últimas novedades en sistemas en seguridad.

El libro incluye algunos programas de los métodos descritos. Estos programas han sido realizados en BASIC de Microsoft y Turbo Pascal. Pueden ejecutarse sobre computadores que dispongan de los sistemas operativos MS-DOS y CP/M. Las versiones de los programas escritos en BASIC han sido realizadas por Agustín Aparicio y la versión del programa DES en Pascal ha sido realizada por Juan Pavón.

En el capítulo de Técnicas Avanzadas se han descrito las fórmulas sobre las que se fundamentan los algoritmos que en él se describen; pese a que estas fórmulas son relativamente complejas, ha sido necesaria su inclusión para aumentar la comprensión sobre estos métodos, para aquellos que quieran profundizar en la materia.

Constantemente hemos tenido que referirnos a los términos congruencia y módulo; por ello vamos a definir estos términos básicos para aquellos que no los recuerden. Se dice que dos o más números son congruentes respecto a un tercero, llamado módulo, cuando divididos uno a uno por el módulo dejan el mismo resto o residuo. En el libro hemos usado normalmente el módulo 26, dado que el alfabeto que hemos utilizado se compone de 26 letras; así el número 6 y el 32 son congruentes respecto al módulo 26, de forma que si dividimos 26 entre 6 nos da de residuo 2, y si dividimos 32 entre 26 también nos produce de residuo 2, por lo que podríamos escribir: 32 = 6(mód. 26), siendo = el símbolo de equivalencia.

INTRODUCCION E HISTORIA 1



LA ciencia que estudia las escrituras secretas se llama criptología; de ella se derivan dos ramas, que son: el criptoanálisis, que intenta descifrar los mensajes cifrados y descubrir así el secreto oculto, y la criptografía.

El término criptografía proviene de los vocablos griegos «cripto», que significa secreto, y «grafia», que significa escritura. Es, pues, una escritura secreta, y consiste en que, partiendo de un mensaje original entendible, se obtiene otro no entendible para un supuesto interceptor; sin embargo, el mensaje resulta comprensible para el destinatario que conoce las reglas de transformación. El proceso de conversión del mensaje original en el otro, incomprensible para el interceptor, se llama cifrado o código y el resultado producido, mensaje cifrado o criptograma. El proceso inverso de obtención del mensaje original a partir del mensaje cifrado se llama descifrado, y el conjunto de reglas que permiten estos procesos se llama clave o llave.

La diferencia entre codificación y cifrado consiste en que, mientras una codificación sustituye una palabra o frase codificada por una palabra o frase del texto original, un sistema de cifrado actúa sobre caracteres antes que sobre palabras o frases.

Otra diferencia a resaltar es la existencia de los sistemas de algoritmos o clave secreta, frente a los de clave pública. Los primeros no son conocidos más que por los que los utilizan, permaneciendo en secreto el método. En los segundos se conoce cómo opera el sistema permaneciendo sólo oculta la clave que cada vez se utiliza.

Aunque parezca que los algoritmos de clave pública serían más fáciles de descubrir que los de clave secreta, en realidad no es así, pues los de clave pública están más contrastados y si mostraran alguna debilidad rápidamente se daría a conocer y sería desechado.

Desde que el hombre contó con la escritura como vehículo de comunicación, se hizo necesario un cuidado especial para imposibilitar la lectura de información con carácter privado.

Los sistemas más elementales utilizados al principio para enviar mensajes privados fueron recipientes cerrados conteniendo el mensaje, pero para obtener dicho mensaje valía con capturar al mensajero; esto hizo necesario ocultar de algún modo el significado del mensaje, para que su captura no supusiera su interpretación.

En el Antiguo Egipto había dos tipos de escritura: una que usaba la gente del pueblo llamada demótica, y otra, denominada hierática, que solo era conocida por los sacerdotes.

Los lacedemonios crearon un sistema de criptografía que consistía en escribir longitudinalmente sobre una tira de pergamino que había sido enrollada sobre un bastón de grosor y longitud determinada; el receptor debía poseer un bastón de las mismas características que el emisor, para así poder enrollar la tira de pergamino recibida y que las letras al leerse tomaran el sentido del texto claro; estos bastones se llamaban escítalos.



Escitala

Carlomagno hizo uso de un sistema de sustitución; éste consistía en que a cada letra se le asignaba un símbolo; su alfabeto con sus equivalencias es el siguiente:

a b c d e f g h i j k l m n o p q r s t u v w x y z

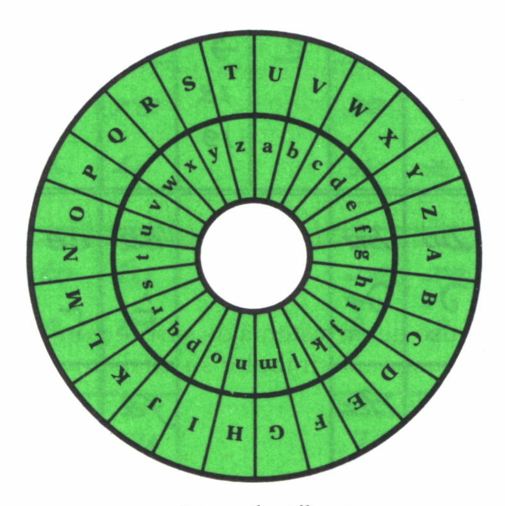
Un griego llamado Histiaeus, que en una ocasión precisó enviar un mensaje a su yerno Aristágoras para que capitaneara una revuelta, se valió de un ingenioso método: hizo afeitar la cabeza a un esclavo, escribió en ella el mensaje, esperó a que le creciera de nuevo el cabello hasta ocultar el mensaje y lo envió a su yerno; éste no tuvo más que afeitar la cabeza del esclavo para poder leer así el mensaje.

En el siglo I a. C. César utilizó para comunicarse con sus generales un método que consistía en sustituir cada letra del mensaje por otra seleccionada de una forma fija; esta sustitución se hacía dentro de las letras del propio alfabeto, por lo que este método se denomina de transposición.

Este método y otros similares tienen una facilidad para su descifrado, y es la «huella dactilar» que cada lengua posee; todas las lenguas tienen unas especiales características de regularidad (más adelante veremos este punto).

En los siglos siguientes no se tiene noticia del desarrollo de nuevos métodos de criptografía, hasta llegar a la Edad Media, donde algunas Repúblicas italianas descubrieron un fácil sistema para evitar el análisis de frecuencias: la sustitución de una misma letra por varios signos diferentes, incluyendo el añadido de signos sin valor a los espacios entre palabras; muestra de estas técnicas es la que empleaba San Bernardino en sus viajes evangelizadores: un signo para cada consonante, tres distintos para cada vocal, seis sin valor y diez figuras como nombres-código para los propios.

La primera máquina de criptografiar fue producida por León Battista Alberti en Roma, durante el Renacimiento. Consiste en dos discos concéntricos con el mismo eje, en cada disco hay un alfabeto y girando uno de ellos se obtiene un alfabeto de transposición.



Disco de Alberti

Cada vez que se envíe un mensaje, se utiliza un nuevo alfabeto, por lo que sólo se repetirá el mismo alfabeto cada 26 mensajes.

En el siglo XVI, Girolamo Cardano, nacido en Pavía, diseñó un sistema de criptografía, consistente en una carta con agujeros perforados, que había que colocar, para poderla interpretar, sobre un determinado texto preestablecido.

En este mismo siglo, en España, se conoce la cifra usada por Felipe II, que al subir al trono cambió la clave usada desde Carlos V. A continuación se muestra la cifra usada por este rey español.

a ɔ 10 11	b ∂	c r	d σ	e tt 12 13	f a	g 2	h v	i n 14 15	l ϙ	m ɤ	n 9
o ɵ 16 17	p ω	q ∧	r 4	s æ	t e	u ∋ 18 19	x σ'	y —o	z ƀ		

ba	be	bi	bo	bu	ca	ce	ci	co	cu
da	de	di	do	du	fa	fe	fi	fo	fu
ga	ge	gi	go	gu	ha	he	hi	ho	hu
ja	je	ji	jo	ju	la	le	li	lo	lu

ma	me	mi	mo	mu	na	ne	ni	no	nu
pa	pe	pi	po	pu	qua	que	qui	quo	quu
ra	re	ri	ro	ru	sa	se	si	so	su
ta	te	ti	to	tu	xa	xe	xi	xo	xu
ya	ye	yo	yi	yu	za	ze	zi	zo	zu
bla	ble	bli	blo	blu	bra	bre	bri	bro	bru
cha	che	chi	cho	chu	cla	cle	cli	clo	clu

cra	cre	cri	cro	cru		dra	dre	dri	dro	dru
F́	F.	Ḟ·	F.	Fe		ǵ	g.	ġ	g·	ge
fla	fle	fli	flo	flu		fra	fre	fri	fro	fru
h́	ḥ	ḣ	h.	he		H́	Ḥ	Ḣ	H.	He
gla	gle	gli	glo	glu		gra	gre	gri	gro	gru
ṕ	p.	ṗ	p.	pe		Ṕ	P.	Ṗ	P.	Pe
pla	ple	pli	plo	plu		pra	pre	pri	pro	pru
q́	q̇	.q	q.	qe		2́	2.	2̇	2.	2e
tra	tre	tri	tro	tru						
Ŕ	Ṛ	Ṙ	R.	Re						

— A —		Aviers —	meo	Avinon —	tia	Balduino —	ri
Alemania —	cr	Avisso —	qui	— B —		Burdeos —	um
Alemanes —	rat	Agora —	dem	Brabante —	qui	Bayona —	cre
Armas —	lon	Aunque —	sen	Bredereda —	im	Bruxelas —	dat
Artilleria —	ge	Alminante —	tén	Bastimento —	pe	Borbon —	gra

— C —		— D —		Lorena —	por	Francia —	22
Consejo —	ui	Dios —	iun	Duque de	test	Frances —	23
Catholico —	us	Duque —	gi	Omala		Frontera —	24
Cardenal —	aut	Duquesa —	tur	Duque de	quid	— G —	
Chanciller —	sta	Designo —	ne	Vandoma		Gente —	25
Chatillon —	bi	Despacho —	que	— E —		Guerra —	26
Conde —	lius	Dinero —	sal	Emperador —	nam	Gobernador	28
Christian.mo —	es	Diligencia —	sum	España —	ubi	General —	27
Christiano —	fe	Duque de Anjou	pror	Españoles —	am	Govierno —	29
Campo —	vi	D. Frances de	sus	Embaxador	or	Guarnicion —	30
Cargo —	quod	Alava		Embaxada —	non	Gasto —	31
Coronel —	fit	Duque de Ne-	neg	Escocia —	in	Grande —	32
Capitan —	quam	murs		Egmont —	est	Gante —	33
Caualleros —	il	Duque de Ne-	su	Estado —	et	Grisones —	34
Caualleria —	lud	vers		Exercito —	adm	— H —	
Correo —	ꝑ	Duque de Mon-	ave	Effecto —	15	Hombre —	35
Carta —	am	pensier		Espia —	celur	Heiege —	36
Cosa —	ci	Duque de	esse	— F —		Herremelo —	37
Como —	tia	Guisa		Flandes —	20	— I —	
Conçierto —	ad	Duque de		Flamenco —	21	Imperio —	38

Italia —	39	Ministro —	tum	— P —		Rey —	79
Inglaterra —	40	Moulac —	id	Papa —	63	Reyno —	81
Ingleses —	41	Montigni —	mel	Principe —	64	Republica —	82
Infantes —	42	Mas —	fa	Provincia —	65	Remedio —	83
Infanteria —	43	Menos —	ciet	Persona —	66	Respuesta —	84
Inquisicion —	44	— N —		Provision —	67	Resolucion —	85
Intelligencia —	45	Negocio —	51	Porque —	68	Raistres —	86
Importancia —	46	Necesidad —	52	Para —	69	Reyna —	80
— L —		Navio —	53	Paraque —	70	— S —	
Lucemburg —	47	Nuncio —	54	Pero —	71	Su Mag.d —	87
Lutherano —	48	Noncarme —	55	Puerto —	72	Su Alt.a —	88
Liga —	49	Navarra —	56	— Q —		Su Exc.a —	89
Libertad —	50	Nunca —	57	Quando —	:	Savoya —	90
Lorena —	sed	— O —		Quanto —	74	Suyza —	91
Licencia —	qu	Obispo —	58	Qualidad —	75	Suyzos —	92
Luego —	od	Ocasion —	59	Qu[illegible]ntidad —	7[illegible]	Señor —	93
— M —		Orden —	60	Qual —	77	Servicio —	94
Memoransi —	est	Officio —	61	Quan —	78	Secretario —	95
Mompensier —	ama	Oranges —	62	— R —		Socorro —	96

Sueño	97	Tratado	99	V. Mag.d	cre	Villa	fri
Siempre	98	Tener	bli	V. Ex.a	cri	Visorey	fro
— T —		Tanto	blo	V. S.a	cro	Vencer	fru
Trato	98	Todo	blu	V. m.	cru	Ugenote	gral
Tierra	bla	Toda	cra	Vuestro	fra		
Tiempo	ble	— V —		Vuestra	fre		

Nulas seran todas las letras dictiones o numeros despues de los quales se siguiese una ·S· entre dos puntos y todo el renglon que començare en una :N· entre dos puntos :ó parte del hasta topar una +.

En estos cuadros vemos que las vocales a, e, i y o pueden ser sustituidas por varias letras, números o trazos especiales, diferentemente: así, la frase «vamos a la guerra» podría cifrarse de las tres formas siguientes, y de algunas más:

e ∂ ɔ- ∀ ӿ 10 ♀́ 2̄6

e 10 ɔ- 16 ӿ 11 ♀́ 2̄6

e 11 ɔ- 17 ӿ ∂ ♀́ 2̄6

Aunque para aquel tiempo parezca un método bastante seguro, el francés Viete consiguió descifrarlo y comunicárselo a su rey Enrique IV de Francia, hecho este que llegó a oídos de Felipe II, quien creyendo su código indescifrable, trató de que el Vaticano juzgara a Viete por usar magia negra para descifrarlo.

En el siglo XVII en la época de Carlos I de Inglaterra, se utilizaban métodos de codificación silábica y sustitución.

En el siglo XIX Napoleón utilizó varios sistemas de cifrado, basados en el sistema denominado Richelieu y Rossignol. También utilizó el sistema de asignación de símbolos numéricos a grupos de una o más letras.

Con el uso de las señales de telegrafía (puntos y rayas), el norteamericano Verman diseñó un sistema de criptografía que consistía en asociar al punto un cero y a la raya un uno; así, la palabra CRIPTO en código Morse se escribe -.-./.-./../.--./-/---/, correspondiendo los / a la separación entre letras; este código, pasado a 0 y 1, sería:

1010/010/00/0110/1/111/

Si se envía esta secuencia de dígitos, ya no es posible detectar fácilmente la «huella dactilar» del lenguaje, pero el código Morse es bien conocido y, por tanto, fácilmente interpretable; para resolver este problema, Verman introdujo, por primera vez en la historia de la criptografía, el azar; éste consistía en lanzar una moneda al aire. Si salía cara, representaba el 0, y cruz el 1. Tiraba tantas veces la moneda como letras tuviera el mensaje, con la siguiente clave:

0 + 0 = 1, 1 + 0 = 1 y 1 + 1 = 0

Así, la palabra CRIPTO, suponiendo que las seis tiradas de la moneda hubieran sido: cara, cruz, cruz, cara, cruz y cruz, sería enviada como:

1111/101/11/1111/0/000

lo que difícilmente podría ser descifrado por los interceptores del mensaje, pues, como vemos, en el mensaje a enviar la primera y la cuarta letra coinciden, correspondiendo a dos letras diferentes la C y la P del mensaje original, pero este sistema es complejo y lento para sus utilizadores.

Como en muchos otros campos de la Ciencia, las dos Guerras Mundiales impulsaron la criptografía hasta diseños muy sofisticados.

Durante la Primera Guerra Mundial los alemanes utilizaron el sistema denominado A D F G X, para el cifrado de mensajes a través de la siguiente matriz:

	A	D	F	G	X
A	n	b	x	r	u
D	q	o	k	d	v
F	a	h	s	g	f
G	m	z	c	l	t
X	e	i	p	j	w

Así la palabra SECRETO se convertía en FFXAGFAGXAGXDD; el sistema se completaba con una transposición de bloques con longuitud 20. Con posterioridad se añadió a la matriz la letra V, dando el sistema ADFGVX, con lo que se podían tratar también los números.

	A	D	F	G	V	X
A	n	b	x	r	u	1
D	q	o	k	d	v	2
F	a	h	s	g	f	3
G	m	z	c	l	t	4
V	e	i	p	j	w	5
X	6	7	8	9	0	

Una variante del disco de Alberti fue diseñado por el presidente norteamericano Jefferson; su sistema consistía en varios discos formando un cilindro. En cada disco estaba escrito el alfabeto completo; al entrar por un extremo una señal se permuta al paso de cada disco, el contacto entre cada disco se realiza mediante un cableado. Esto, indudablemente, produce siempre las mismas permutaciones; para que esto no pase, los discos se pueden mover y así cambiar de un mensaje a otro.

a	j	c	h	u	z	l	r	t	b	x
b	k	d	i	v	a	m	s	u	c	y
c	l	e	j	w	b	n	t	v	d	z
d	m	f	k	x	c	o	u	w	e	a

Tambor de Jefferson

La máquina Hagelin C-48 fue construida en 1940 y consiste en seis volantes unidos por un eje, donde los volantes tenían 26, 25, 23, 21, 19 y 17 dientes, respectivamente; por una ventana se puede ver en cada momento el diente de cada volante. Al comenzar el proceso se vería AAAAAA, después de 17 giros se vería RRRRRA; dado que el último disco ha terminado su rotación, solamente el 1.er disco contiene el alfabeto completo. Este dispositivo produce la clave que se combina con el texto original.

Otra máquina construida durante la Segunda Guerra Mundial fue la alemana Enigma, que se basaba en un perfecionamiento del cilindro de Jefferson, pero la máquina británica Colossus consiguió descifrar los mensajes que cifraba Enigma.

Un grave error en un sistema de seguridad se produjo durante esta guerra en el departamento de espionaje inglés (SOE); en su campaña en Holanda, este departamento envió a Holanda 60 agentes; todos ellos fueron capturados por los alemanes, los cuales usaron sus claves y transmisores para enviar a Inglaterra mensajes falsos, y el fallo consistió en que el primer agente enviado por el SOE envió sus transmisiones sin el control de seguridad acordado, que era una clave previa a enviar con los mensajes. Este «descuido» supuso que prácticamente la Resistencia holandesa no tuviera efectividad durante esta contienda.

Los estadounidenses construyeron en esta guerra la máquina Magic, que descifraba el código «púrpura» japonés; por este motivo, se consiguió eliminar al almirante japones Yamamoto, en un desplazamiento que realizó en avión, y ello cambió seguramente el curso de la guerra en el Pacífico.

También los estadounidenses hicieron uso de un curioso sistema para sus comunicaciones por radio durante la última Guerra Mundial, y fue el uso de indios navajos para realizar sus transmisiones, pues es casi imposible aprender su idioma si no es dentro de la propia tribu, posibilidad ésta muy remota para los japoneses, contra los que se usó este sistema.

En 1950, con la aparición de los primeros computadores, la criptografía alcanzó su máximo desarrollo. El principio de Verman, es decir, el código binario, ceros y unos, es el que usan los computadores, así se consiguieron eficaces herramientas para criptografiar rápidamente.

METODOS TRADICIONALES 2



XISTEN dos métodos criptográficos tradicionales, que son la transposición y la sustitución, y un tercer método híbrido.

TRANSPOSICION

Consiste en una reagrupación de los caracteres que constituyen el texto del mensaje, de una forma establecida. Es decir, los caracteres cambian su posición pero no su identidad. Los caracteres del mensaje original se toman fuera de su orden en el texto y son reubicados de acuerdo con algún patrón geométrico definido, o camino topológico, convenido «a priori» por los interlocutores válidos. Recibe el nombre de transposición «monoliteral» cuando se efectúa sobre grupos de caracteres del texto claro.

Una transposición elemental consiste en escribir el texto completo, pero en orden inverso, y separado en bloques de cinco caracteres, para que pierda su estructura; así, por ejemplo:

TEXTO: ESTO ES UN MENSAJE

TEXTO CIFRADO: EJASN EMNUS EOTSE

Otra variante sería transmitir cada palabra en orden inverso; así, para el texto anterior:

TEXTO CIFRADO: OTSE SE NU EJASNEM

Dentro de esta categoría de cifrado por transposición existen algunas variantes más sofisticadas, que van a considerarse a continuación.



Sistemas de cifrado por líneas

Se dice que fue utilizado en la guerra civil americana. Es un método simple y puede ser combinado con otros sistemas. Existen dos versiones; en la primera versión, la mitad del texto se escribe en una línea, y la otra mitad debajo. Así:

TEXTO: ESTE ES EL MENSAJE X

PASO 1: ESTEESEL
MENSAJEX

A continuación, se escogen, de arriba abajo, y de izquierda a derecha, bloques de cinco caracteres; en este caso:

TEXTO CIFRADO: EMSET NESEA SJEEL X

En la segunda versión el texto se forma escribiendo el mensaje por columnas, de izquierda a derecha, así:

PASO 1: ETEEMNAE
SESLESJX

Y, a partir de aquí, se escriben las dos filas en bloques de cinco caracteres:

TEXTO CIFRADO: ETEEM NAESE SLESJ X

Una variante a este método es, en vez de escribir el mensaje en columnas, hacerlo en forma de dientes de sierra, y después reagruparlo por filas.

Si la profundidad de los dientes de sierra es 2, este método produce el mismo resultado que el anterior:

```
MENSAJE: E T E E M N A E
          S E S L E S J X
```

TEXTO CIFRADO: ETEEM NAESE SLESJ X

pero si aumentamos la profundidad de los dientes, el texto resultante varía:

```
          E   E   M   A
MENSAJE:   S E S L E S J X
            T   E   N   E
```

TEXTO CIFRADO: EEMAS ESLES JXTEN E

Un programa que resuelve el cifrado-descifrado de dientes de sierra es el siguiente:

```
10 REM DIENTE DE SIERRA
20 CLS:REM <-- EN COMMODORE SUSTITUIR POR: PRI
NT "[SHIFT-HOME]"
30 LOCATE 1,1:PRINT "METODO DE DIENTE DE SIERR
A"
40 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
50 LOCATE 3,1:INPUT "MODO (C/D) :";N$
60 IF N$="F" THEN GOTO 340
65 IF N$<"C" OR N$>"D" THEN GOTO 50
70 LOCATE 5,1:INPUT "CLAVE :";CL
90 IF CL<2 OR CL>99 THEN GOTO 70
100 LOCATE 3,1:PRINT "   MODO = ";N$;"
"
110 LOCATE 5,1:PRINT "  CLAVE = ";CL;"
"
150 LOCATE 7,1:INPUT "MENSAJE : ";M$
160 REM LIMPIA LOS ESPACIOS DEL MENSAJE
170 GOSUB 5000
250 LOCATE 7,1:PRINT "MENSAJE = ";M$;"
"
255 REM NUMERO DE CARACTERES A TRANSPONER
260 LET N=2*(CL-1)
265 REM SELECCION DEL MODO CIFRADO/DESCIFRADO
270 IF N$="C" THEN GOSUB 2000:GOTO 275
272 GOSUB 3000
275 REM FINALIZACION O ITERACION
300 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N) :
 "
310 R$=INKEY$:IF R$="" THEN GOTO 310:REM <-- E
N EL COMMODORE SUSTITUIR POR: GET R$:IF R$="" THEN
 GOTO 310
320 IF R$<>"N" AND R$<>"S" THEN GOTO 310
330 IF R$="S" THEN RUN
340 REM FIN
350 END:REM <-- EN EL SPECTRUM SUSTITUIR POR:
STOP
2000 REM CIFRADO
2010 PRINT "MENSAJE CIFRADO = ";
2020 FOR I=1 TO LM STEP N
2030 PRINT MID$(M$,I,1);
2040 NEXT I
2050 FOR I=1 TO CL-2
2060 FOR J=1 TO LM STEP N
2070 IF J+I<=LM THEN PRINT MID$(M$,J+I,1);
2080 IF J+N-I<=LM THEN PRINT MID$(M$,J+N-I,1);
2090 NEXT J
```

```
52100 NEXT I
2110 FOR I=CL TO LM STEP N
2120 PRINT MID$(M$,I,1);
2130 NEXT I
2140 RETURN
3000 REM DESCIFRADO
3005 PRINT "MENSAJE DESCIFRADO = ";
3010 DIM C$(LM)
3020 LET L=0
3030 FOR I=1 TO LM STEP N
3040 LET L=L+1
3050 LET C$(I)=MID$(M$,L,1)
3060 NEXT I
3070 FOR I=1 TO CL-2
3080 FOR J=1 TO LM STEP N
3090 IF J+I<=LM THEN LET L=L+1:LET C$(I+J)=MID
$(M$,L,1)
3100 IF J+N-I<=LM THEN LET L=L+1:LET C$(J+N-I)
=MID$(M$,L,1)
3110 NEXT J
3120 NEXT I
3130 FOR I=CL TO LM STEP N
3140 LET L=L+1
3150 LET C$(I)=MID$(M$,L,1)
3160 NEXT I
3170 FOR I=1 TO LM:PRINT C$(I);:NEXT I
3180 C$=""
3190 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SE
AN LETRAS DEL MENSAJE
5010 LET I=2
5020 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" T
HEN LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET
I=I-1
5030 LET I=I+1
5040 IF I<=LEN(M$) THEN GOTO 5020
5050 LET LM=LEN(M$)
5070 RETURN
```

VARIACIONES AL PROGRAMA
=======================

COMMODORE:

DEBIDO A LA INEXISTENCIA DE LA SENTENCIA LOCATE EN EL COMMODORE, CADA VEZ QUE ESTA APAREZCA SE SUSTITUIRA POR UNA LLAMADA AL SIGUIENTE PROGRAMA Y DE LA SIGUIENTE FORMA:

SI MIRAMOS LA LINEA 30, EN ESTA PONE:

```
   30 LOCATE 1,1:PRINT "........ETC.
```

LA SUSTITUIREMOS POR:

```
   30 XX=1:YY=1:GOSUB 6000:PRINT "........ETC.
```

Y A PARTIR DE LA LINEA 6000 INTRODUCIREMOS LA SIGUIENTE SUBRUTINA:

```
    6000 PRINT "[HOME]";
    6010 FOR ZZ=1 TO XX:PRINT "[CURSOR DERECHA]
";:NEXT ZZ
    6020 FOR ZZ=1 TO YY:PRINT "[CURSOR ABAJO]";
:NEXT ZZ
    6030 RETURN
```

COMO SE PUEDE APRECIAR, SE MANDA EN LA VARIAB LE XX EL VALOR DE LA COLUMNA A LA QUE SE QUIERE IR , Y EN YY EL VALOR DE LA FILA.

ESTA RUTINA HABRA DE UTILIZARSE EN TODOS LOS PROGRAMAS DE ESTE LIBRO Y DE LA MISMA MANERA.

SPECTRUM:

EN TODOS LOS LUGARES DONDE APAREZCA UN LOCATE SEGUIDO DE UN PRINT, SE PROCEDERA DE LA SIGUIENTE MANERA:

SI POR EJEMPLO TENEMOS LA LINEA 50:

```
   30 LOCATE 5,1:PRINT "......ETC.
```

PONDRIAMOS LO SIGUIENTE:

```
   30 PRINT AT 5,1;"........ETC
```

O LO QUE ES IGUAL, QUITARIAMOS EL LOCATE Y PO NDRIAMOS DESPUES DEL PRINT LA FUNCION AT, SEGUIDA DE LOS NUMEROS DEL LOCATE EN EL MISMO ORDEN Y SEGUIDO DE UN PUNTO Y COMA (;)

LA FUNCION MID$ NO PERTENECE AL BASIC DEL SPE CTRUM POR LO QUE HABRA QUE SUSTITUIRLA. PARA HACER LO FIJEMONOS EN LA LINEA 2030

```
   2030 PRINT MID$(M$,I,1)
```

```
ESTO EN EL SPECTRUM SE PONDRIA COMO:

2030 PRINT M$(I)

O LO QUE ES IGUAL:

   2030 PRINT M$(I TO 1)

ASI PUES SI TUVIESEMOS EL CASO DE:

   1023 PRINT MID$(M$,F+J-1,3)

LO PONDRIAMOS DE LA FORMA:
   1023 PRINT M$(F+J-1 TO 3)
```

```
      ESTO HABRA DE TENERSE EN CUENTA EN TODOS LOS
PROGRAMAS DE ESTE LIBRO Y SE REALIZARA SIEMPRE DE
ESTA MANERA LAS MODIFICACIONES
      NECESARIAS.
```

COMENTARIOS DEL PROGRAMA DIENTE DE SIERRA

Cabecera y mensaje de finalización (20-40). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (50-65). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clave (70-90). Pide la clave del método. Esta clave debe ser un número. Si el número es inferior a 2 o superior a 99, el programa vuelve a pedir la clave.

Imprime modo y clave (100-110). Imprime el modo y la clave para visualizar la elección anterior.

Lectura del mensaje (150). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Selección en función del modo (270). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Tamaño del bloque (260). Se calcula el tamaño de separación entre cada diagonal.

Cifrado (2000-2140). Se imprimen los caracteres del mensaje formando diagonales.

Descifrado (3000-3170). Se imprimen los caracteres del mensaje deshaciendo las diagonales.

Finalización o iteración (300-350). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.



Sistemas de cifrados matriciales

Consisten en escribir el mensaje en una matriz, y a continuación, seleccionar una matriz de forma diferente. Ejemplo:

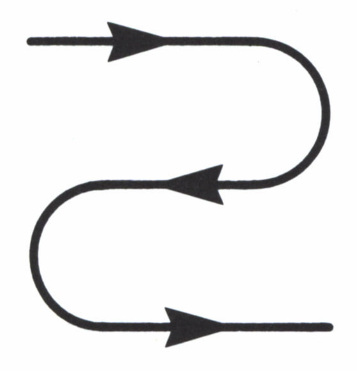
TEXTO: ESTE ES EL MENSAJE X

```
          ESTE
          ESEL
PASO 1:   MENS
          AJEX
```

TEXTO CIFRADO: EEMA SSEJ TENE ELSX

Es decir, el PASO 1 se forma por filas, y el TEXTO CIFRADO por columnas.

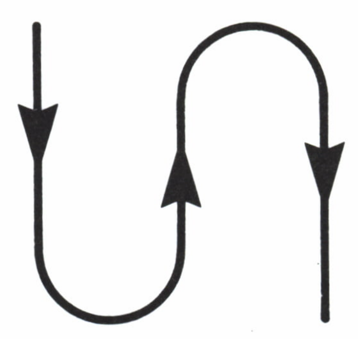
Otro método dentro de este sistema consiste en escribir el texto de la forma:



Así, en nuestro ejemplo anterior sería:

E S T E
L E S E
M E N S
X E J A

y para enviarlo, lo haremos de la forma:



lo que nos daría, escrito en bloques de cinco caracteres:

ELMXE EESTS NJASE E



Sistema de cifrado por columnas a partir de la llave

Sea:

LLAVE = CLAVE

TEXTO = ESTE ES EL MENSAJE X

Primeramente se determina el orden alfabético de los caracteres de la llave; en este caso:

C=2; L=4; A=1; V=5 y E=3

A continuación se escribe el texto en forma matricial, siendo el número de columnas igual al número de caracteres de la llave:

ESTEE
SELME
PASO 1: NSAJE
X

A continuación se escribirá el texto cifrado de acuerdo con la llave. Así, en el ejemplo considerado, primero la tercera columna, ya que la primera letra de la llave es la **A** (posición 3), a continuación la columna 1, ya que la segunda letra de la llave es la **C** (posición 1), etc., quedando agrupados en bloques de cinco caracteres. En consecuencia:

T E E S E
PASO 2: L S E E M
A N E S J
X

y ahora formamos el mensaje, leyendo en orden continuado las columnas:

TEXTO CIFRADO: TLAES NXEEE SESEM J

Un programa que resuelve el cifrado-descifrado del método de columnas es el siguiente:

```
    10 REM TRANSPOSICION DE COLUMNAS
    20 CLS:REM <-- EN COMMODORE SUSTITUIR POR: PRI
NT "[SHIFT-HOME]"
    30 LOCATE 1,1:PRINT "METODO DE TRANSPOSICION D
E COLUMNAS"
    40 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
    50 LOCATE 3,10:INPUT "MODO (C/D) : ";N$
    60 IF N$="F" THEN GOTO 590
    65 IF N$<"C" OR N$>"D" THEN GOTO 50
    70 LOCATE 5,1:INPUT "CLAVE : ";C$
    120 LOCATE 3,1:PRINT "      MODO : ";N$;"
  "
    130 LOCATE 5,1:PRINT "     CLAVE : ";C$;"
 "
    135 REM TRANSFORMAR LA CLAVE
    136 LC=LEN(C$)
    137 GOSUB 1000
    140 REM
    270 LOCATE 7,1:INPUT "MENSAJE : ";M$
    275 REM LIMPIA LOS ESPACIOS DEL MENSAJE
    280 GOSUB 5000
    290 REM PREPARACION DEL MENSAJE
    300 GOSUB 6000
    310 LOCATE 7,1:PRINT "  MENSAJE : ";M$;"
   "
    320 REM SELECION DEL MODO CIFRADO DESCIFRADO
    330 IF N$="C" THEN GOSUB 2000:GOTO 340
    335 GOSUB 3000
    340 REM
    350 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N)
           "
    360 LET R$=INKEY$:IF R$="" THEN GOTO 360:REM <
-- EN EL COMMODORE SUSTITUIR POR: GET R$:IF R$=""
THEN GOTO 360
    370 IF R$<>"N" AND R$<>"S" THEN GOTO 360
    380 IF R$="S" THEN GOTO 20
    390 REM FIN
    400 END:REM <-- EN EL SPECTRUM SUSTITUIR POR:
STOP
    1000 REM TRANSFORMACION DE LA CLAVE
```

```
     1010 LET TM=ASC("]"):REM<-- EN EL SPECTRUM SUS
TITUIR POR: LET TM=CODE("]")
     1020 DIM M(LC)
     1030 FOR I=1 TO LC
     1040 LET T=TM
     1050 FOR J=1 TO LC
     1055 LET L=ASC(MID$(C$,J,1)):REM <-- EN EL SPE
CTRUM CAMBIARLO POR: LET L=CODE(C$(J))
     1060 IF L<T THEN LET T=L:LET P=J
     1090 NEXT J
     1100 MID$(C$,P,1)="]":REM <-- EN EL SPECTRUM S
USTITUIR POR: LET C$(P)="]"
     1110 LET M(I)=P
     1120 NEXT I
     1130 RETURN
     2000 REM CIFRADO
     2010 PRINT "MENSAJE CIFRADO = ";
     2020 FOR I=1 TO LC
     2030 FOR J=M(I) TO LM STEP LC
     2040 PRINT MID$(M$,J,1);:REM <-- EN EL SPECTRU
M SUSTITUIR POR: PRINT M$(J);
     2050 NEXT J
     2060 NEXT I
     2070 RETURN
     3000 REM DESCIFRADO
     3010 DIM C$(LM)
     3020 PRINT "MENSAJE DESCIFRADO = ";
     3030 LET L=0
     3040 FOR I=1 TO LC
     3050 FOR J=M(I) TO LM STEP LC
     3060 LET L=L+1
     3070 LET C$(J)=MID$(M$,L,1):REM <-- EN EL SPEC
TRUM CAMBIARLO POR: LET C$(J)=M$(L)
     3080 NEXT J
     3090 NEXT I
     3100 FOR I=0 TO LM-1:PRINT C$(I);:NEXT I
     3110 RETURN
     5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SE
AN LETRAS DEL MENSAJE
     5010 LET I=2
     5020 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" T
HEN LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET
I=I-1
     5030 LET I=I+1
     5040 IF I<=LEN(M$) THEN 5020
     5060 LET LM=LEN(M$)
     5070 RETURN
     6000 REM RELLENA EL MENSAJE CON "W"
```

```
6020 M$=M$+"W":LM=LEN(M$)
6030 IF LM/LC<>INT(LM/LC) THEN GOTO 6020
6050 RETURN
```

COMENTARIOS DEL PROGRAMA COLUMNAS

Cabecera y mensaje de finalización (20-40). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (50-65). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clave (70-90). Pide la clave del método. Esta clave debe ser una cadena de caracteres.

Imprime modo y clave (120-130). Imprime el modo y la clave para visualizar la elección anterior.

Transformación de la clave (1000-1130). Construye la guía, poniendo en cada elemento de una matriz el orden alfabético de cada letra de la clave.

Lectura del mensaje (270). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Selección en función del modo (330). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2070). Se imprimen los caracteres del mensaje según el orden establecido por la clave.

Descifrado (3000-3110). Se construyen los caracteres del mensaje descifrado según el orden establecido por la clave y a continuación se imprime.

Finalización o iteración (350-400). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.



Sistema de cifrado Cardano

Ya comentamos en la Introducción que en el siglo XVI Girolamo Cardano hizo uso de un sistema, consistente en colocar una cartulina perforada sobre un texto preestablecido; una variante de este método es el siguiente:

Se diseña una cartulina cuadrada sobre la que se perforan un número de agujeros, igual a 1/4 del número total de cuadros de la cartulina y de forma que no se solapen al girar sobre su eje; supongamos que diseñamos

una cartulina con 36 cuadros, por lo que tendremos que perforar 9 agujeros:

X		X		X	X
X		X		X	X
X	X	X	X	X	X
	X		X	X	X
X	X	X		X	
	X	X	X	X	X

colocamos la cartulina sobre un papel y comenzamos a escribir el mensaje, letra a letra, en los agujeros de la cartulina, empezando por la primera fila hasta rellenar por filas todos los agujeros. Se rota la cartulina sobre su centro y así se sigue hasta completar las cuatro posibles rotaciones.

Supongamos que el mensaje a enviar es:

«LLEGARE EL LUNES Y VOY SOLO Y DESARMADO»

En la primera posición de la cartulina tenemos:

	L		L		
	E		G		
A		R			
			E		E
L					

El segundo giro nos produce:

L	L	U	L		
	E		G	N	E
		S			
A	Y	R		V	O
			E		E
L	Y				

El tercer giro nos produce:

L	L	U	L		S
O	E	L	G	N	E
		S	O		Y
A	Y	R		V	O
		D	E	E	E
L	Y	S		A	

El cuarto giro nos produce (se rellenan los huecos que no se cubran con el mensaje, con una letra, por ejemplo, la W).

L	L	U	L	R	S
O	E	L	G	N	E
M	A	S	O	D	Y
A	Y	R	O	V	O
W	W	D	E	E	E
L	Y	S	W	A	W

Ahora leemos el mensaje a transmitir, fila a fila, y en bloques de cinco caracteres,

TEXTO CIFRADO: LLULR SOELG NEMAS ODYAY ROVOW WDEEL LYSWA W

El siguiente programa realiza el cifrado-descifrado de este método de cifrado Cardano.

```
10 REM CARDANO
14 DIM C$(6,6)
15 LET NH=0
16 LET CN=0
19 LET RF=1
20 CLS:REM <-- EN EL COMODORE SUSTITUIR POR: P
RINT "[SHIFT HOME]"
30 LOCATE 1,1:PRINT "METODO DE CIFRADO Y DESCI
```

```
FRADO CARDANO"
    35 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
    40 LOCATE 3,1:INPUT "MODO (C/D) : ";N$
    60 IF N$="F" THEN GOTO 900
    70 IF N$<"C" OR N$>"D" THEN GOTO 40
    90 LOCATE 3,1:PRINT "      MODO = ";N$;"
 "
    100 REM INICIALIZA EL TABLERO
    105 GOSUB 1000
    650 LOCATE 15,1:INPUT "MENSAJE : ";M$
    670 REM LIMPIA LOS ESPACIOS DEL MENSAJE
    680 GOSUB 5000
    700 REM RELLENA EL MENSAJE CON W HASTA QUE SEA
 UN MULTIPLO DE 36
    780 LET M$=M$+"W"
    790 LET LM=LEN(M$)
    800 IF LM/36<>INT(LM/36) THEN GOTO 780
    820 LOCATE 15,1:PRINT "MENSAJE = ";M$
    860 IF N$="C" THEN GOSUB 2000:GOTO 865
    862 GOSUB 3000
    865 LOCATE 10,1:PRINT "DESEA CONTINUAR (S/N)
        "
    870 LET R$=INKEY$:IF R$="" THEN GOTO 870:REM <
-- EN EL COMMODORE SUSTITUIR POR: GET R$:IF R$=""
THEN GOTO 870
    875 IF R$<>"N" AND R$<>"S" THEN GOTO 870
    880 IF R$="S" THEN RUN
    900 REM FIN
    910 END:REM <-- EN EL SPECTRUM SUSTITUIR POR:
STOP
    1000 REM LECTURA DEL TABLERO
    1110 LOCATE 20,1:PRINT "PARA CAMBIAR DE CUADRA
NTE PULSE *      "
    1190 FOR C=1 TO 4
    1200 LET SY=5
    1210 IF C>2 THEN LET SY=9
    1220 LET SX=40:REM <-- EN EL SPECTRUM Y COMMOD
ORE CAMBIAR EL 40 POR: 25
    1230 IF C=2 OR C=3 THEN LET SX=45:REM <-- EN E
L SPECTRUM Y COMMODORE CAMBIAR EL 45 POR: 30
    1240 LOCATE SY+1,SX:PRINT "123"
    1242 LOCATE SY+2,SX:PRINT "456"
    1244 LOCATE SY+3,SX:PRINT "789"
    1250 LOCATE 5,1:PRINT "CUADRANTE ";C;" AGUJERO
"
    1270 LET R$=INKEY$:IF R$="" THEN GOTO 1270:REM
 <-- EN EL COMMODORE SUSTITUIR POR: GET R$:IF R$="
" THEN GOTO 1270
    1300 IF R$="*" THEN GOTO 1620
```

```
1310 IF R$<"1" OR R$>"9" THEN GOTO 1270
1350 LET V=VAL(R$)
1360 LOCATE 5,20:PRINT V
1370 LET GY=INT((V-1)/3)+1
1380 LET GX=V-3*(GY-1)
1390 IF C>2 THEN LET GY=GY+3
1400 IF C=2 OR C=3 THEN LET GX=GX+3
1410 IF G(GY,GX)=1 THEN GOTO 1270
1420 FOR Y=1 TO 6
1430 FOR X=1 TO 6
1440 IF G(Y,X)=0 THEN GOTO 1530
1450 LET YY=Y
1460 LET XX=X
1470 FOR R=1 TO C-1
1480 LET I=YY
1490 LET YY=XX
1500 LET XX=7-I
1510 IF YY=GY AND XX=GX THEN GOTO 1270
1520 NEXT R
1530 NEXT X
1540 NEXT Y
1550 LET G(GY,GX)=1
1560 IF GX>3 THEN LET GX=GX+1
1570 IF GY>3 THEN LET GY=GY+1
1580 LOCATE 5+GY,31:PRINT CHR$(254):REM <-- EN
 EL SPECTRUM SUSTITUIR EL CHR$(254) POR:
1590 LET NH=NH+1
1600 IF NH>=9 THEN LET C=4:GOTO 1620
1610 GOTO 1270
1620 IF C=4 AND NH<9 THEN GOTO 1270
1630 NEXT C
1640 RETURN
2000 REM CIFRADO
2005 PRINT "MENSAJE CIFRADO = ";
2010 FOR C=1 TO 4
2020 FOR Y=1 TO 6
2030 FOR X=1 TO 6
2040 IF G(Y,X)<>0 THEN LET CN=CN+1:LET C$(Y,X)
=MID$(M$,CN,1)
2070 NEXT X
2080 NEXT Y
2090 GOSUB 4000
2100 NEXT C
2110 FOR Y=1 TO 6
2120 FOR X=1 TO 6
2130 PRINT C$(Y,X);
2140 NEXT X
2150 NEXT Y
```

```
2160 LET M$=MID$(M$,37,LEN(M$))
2170 LET LM=LM-36
2175 IF LM>=56 THEN GOTO 2010
2180 RETURN
3000 REM DESCIFRADO
3010 PRINT "MENSAJE DESCIFRADO = ";
3020 FOR Y=1 TO 6
3030 FOR X=1 TO 6
3040 LET CN=CN+1
3050 LET C$(Y,X)=MID$(M$,CN,1)
3060 NEXT X
3070 NEXT Y
3080 FOR C=1 TO 4
3090 FOR Y=1 TO 6
3100 FOR X=1 TO 6
3110 IF G(Y,X)>0 THEN PRINT C$(Y,X);
3120 NEXT X
3130 NEXT Y
3140 GOSUB 4000
3150 NEXT C
3160 LET M$=MID$(M$,37,LEN(M$))
3170 LET LM=LM-36
3180 IF LM>=36 THEN GOTO 3020
3190 RETURN
4000 REM GIRO DE LA PLANTILLA
4010 FOR Y=1 TO 6
4020 FOR X=1 TO 6
4030 IF G(Y,X)=RF THEN LET G(Y,X)=0:LET YY=X:L
ET XX=7-Y:LET G(YY,XX)=3-RF
4040 NEXT X
4050 NEXT Y
4060 LET RF=3-RF
4070 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SE
AN LETRAS DEL MENSAJE
5010 LET I=2
5020 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" T
HEN LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET
I=I-1
5030 LET I=I+1
5040 IF I<LEN(M$) THEN GOTO 5020
5050 LET LM=LEN(M$)
5060 RETURN
```

COMENTARIOS DEL PROGRAMA CARDANO

Inicialización (10-20). Se inicializan algunas variables y las tablas que empleará el programa.

Cabecera y mensaje de finalización (30-35). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (40-70). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Imprime modo (90). Imprime el modo para visualizar la elección anterior.

Inicialización del tablero (1000-1600). Dibuja una plantilla que servirá para introducir los huecos de cada cuadrante. Cada hueco se introduce como número, comprobando que no sea inferior a 1 ni superior a 9; si el hueco ya estaba o al girar la plantilla está ocupado, el programa vuelve a pedir el hueco.

Lectura del mensaje (650). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Prepara mensaje (6000-6040). Rellena el mensaje con «W» hasta que la longitud del mensaje sea un múltiplo de 36.

Selección en función del modo (860). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2180). Para cada uno de los caracteres del mensaje se le cifra con un cuadrante, y a continuación se gira la plantilla para el siguiente cuadrante. Cuando finaliza la construcción del mensaje cifrado, se imprime.

Descifrado (3000-3190). Se construye el mensaje en forma de matriz. Se cifra con un cuadrante, se imprime y a continuación se gira la plantilla para el siguiente cuadrante.

Finalización o iteración (865-910). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Los métodos de transposición se pueden aplicar varias veces, de forma que al texto ya cifrado se le puede aplicar una nueva transposición.



SUSTITUCION

Consiste en el reemplazamiento de los caracteres del texto por otros caracteres. Los caracteres mantienen su posición, pero pierden su identidad. Un ejemplo es el Sistema «César», ya comentado en la Introducción, que consiste en sustituir cada letra por la letra colocada tres lugares a la derecha en el alfabeto. Así:

ALFABETO: ABCDEFGHIJKLMNOPQRSTUVWXYZ
ALFABETO CIFRADO: DEFGHIJKLMNOPQRSTUVWXYZABC

Así, por ejemplo,

TEXTO: ESTO ES UNA PRUEBA

TEXTO CIFRADO: HVWR HV XQD SUXHED

Una variación del método César es el de orden inverso al normal, así:

ALFABETO: ABCDEFGHIJKLMNOPQRSTUVWXYZ

ALFABETO CIFRADO: WVUTSRQPONMLKJIHGFEDCBAZYX

para nuestro ejemplo:

TEXTO: ESTO ES UNA PRUEBA

TEXTO CIFRADO: SEDI SE CJW HFCSVW

Otra variante es el uso de un alfabeto recíproco, ya que aplicado dos veces, devuelve el carácter original:

ALFABETO: ABCDEFGHIJKLMNOPQRSTUVWXYZ

ALFABETO CIFRADO: LKJIHGFEDCBAZYXWVUTSRQPONM

Un programa que realiza estos cifrados y descifrados es el siguiente:

```
10 REM METODO CESAR DE CIFRADO Y DESCIFRADO
20 CLS: REM <-- EN EL COMMODORE CAMBIARLO POR: PRI
NT "[SHIFT-HOME]"
30 LOCATE 1,1:PRINT "METODO CESAR DE CIFRADO Y DES
CIFRADO"
40 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
50 LOCATE 3,1:INPUT "MODO (C/D) : ";N$
60 IF N$<"C" OR N$>"D" THEN GOTO 50
80 LOCATE 5,1:INPUT "CLAVE : ";C$
90 IF C$<"A" OR C$>"Z" THEN GOTO 80
100 IF LEN(C$)>1 THEN GOTO 80
110 LOCATE 3,1:PRINT "     MODO = ";N$
120 LOCATE 5,1:PRINT "    CLAVE = ";C$
130 LET CL=ASC(C$)-65:REM <-- EN EL SPECTRUM CAMBI
AR POR: LET CL=CODE(A$)-65
140 LOCATE 7,1:INPUT "MENSAJE : ";M$
145 REM
150 IF N$="D" THEN LET CL=-CL:PRINT "MENSAJE DESCI
FRADO = ";:GOTO 160
155 PRINT "MENSAJE CIFRADO = ";
160 GOSUB 1000
```

```
230 REM
240 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N)
   "
250 LET R$=INKEY$:IF R$="" THEN GOTO 250:REM <-- E
N EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" THEN
 GOTO 250
260 IF R$<>"S" AND R$<>"N" THEN GOTO 250
270 IF R$="S" THEN RUN
280 REM FIN
290 END:REM <-- EN EL SPECTRUM CAMBIAR POR: STOP
1000 REM PONE ESPACIOS EN BLANCO EN LOS CARACTERES
 NO VALIDOS
1005 LET LM=LEN(M$)
1010 FOR I=1 TO LM
1020 LET T$=MID$(M$,I,1):REM <-- EN EL SPECTRUM CA
MBIARLO POR: LET T$=M$(I)
1030 IF T$<"A" OR T$>"Z" THEN PRINT " ";:GOTO 1040
1035 GOSUB 2000
1040 NEXT I
1050 RETURN
2000 REM ESTA RUTINA REALIZA EL DESPLAZAMIENTO DE
LA CLAVE
2001 REM EN CADA CARACTER DEL MENSAJE Y LO IMPRIME
2010 LET C=ASC(T$)-65+CL:REM <-- EN EL SPECTRUM CA
MBIARLO POR: LET C=CODE(T$)-65+CL
2030 IF C<0 THEN LET C=C+26
2040 IF C>=26 THEN LET C=C-26
2050 PRINT CHR$(C+ASC("A"));:REM <-- EN EL SPECTRU
M CAMBIARLO POR: PRINT CHR$(C+CODE("A"));
2060 RETURN
```

COMENTARIOS DEL PROGRAMA CESAR

Cabecera y mensaje de finalización (10-40). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (50-70). Pide el modo de funcionamiento del programa; este puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clave (80-100). Pide la clave del método. Esta clave debe ser una letra del alfabeto (A..Z). Si la clave introducida no es una letra o tiene una longitud mayor que uno, el programa vuelve a pedir la clave.

Imprime modo y clave (110-120). Imprime el modo y la clave para visualizar la elección anterior.

Lectura del mensaje (140). Pide el mensaje para descifrar o cifrar.

Selección en función del modo (150). Si el modo es «D» (descifrado), cambia de signo la clave e imprime «MENSAJE DESCIFRADO = ». Si el modo es «C» (cifrado), imprime «MENSAJE CIFRADO = ».

Limpia mensaje y desplaza en función de la clave (1000-2060). Los caracteres que no pertecen al alfabeto no se consideran válidos y en su lugar se imprime un espacio; al resto de los caracteres se les suma la clave y a continuación se imprimen.

Finalización o iteración (230-290). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Otra variante de este método César es la conversión intermedia de las letras en números, siguiendo el siguiente procedimiento:

1. Reemplazar cada letra del alfabeto por el número natural en orden consecutivo desde el 1 en adelante.
2. Multiplicar los números resultantes del paso 1 por un número que se prefije; si esta multiplicación excede de 26, se reemplaza por el residuo equivalente.
3. Sustituir cada número resultante por su letra equivalente, con lo que obtendremos el alfabeto codificado.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
P.1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P.2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52
RES.	2	4	6	8	10	12	14	16	18	20	22	24	26	2	4	6	8	10	12	14	16	18	20	22	24	26
P.3	B	D	F	H	J	L	N	P	R	T	V	X	Z	B	D	F	H	J	L	N	P	R	T	V	X	Z

Por lo que el alfabeto con su sustitución es el siguiente:

ALFABETO:
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ALFABETO CIFRADO:
B D F H J L N P R T V X Z B D F H J L N P R T V X Z

Así, dado el mensaje:

	H	A	Y	Q	U	E	E	S	P	E	R	A	R
PASO 1	8	1	25	17	21	5	5	19	16	5	18	1	18
PASO 2	16	2	50	34	42	10	10	38	32	10	36	2	36
*2													
RESIDUO	16	2	24	8	16	10	10	12	6	10	10	2	10
PASO 3	P	B	X	H	P	J	J	L	F	J	J	B	J

por lo que enviaríamos:

PBXHPJJLFJJBJ

Más complicado es el método siguiente, que opera como se indica:

1. Reemplazar cada letra del alfabeto por el número que le corresponda, A=1, B=2, etc.
2. Sumar a los números resultantes del paso 1 un número.
3. Multiplicar los números resultantes del paso 2 por un número; si esta multiplicación excede de 26, se reemplaza por el residuo.
4. Sustituir el número resultante por la letra equivalente, con lo que se obtiene el alfabeto cifrado.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
P.1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P.2 +3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	1	2	3
P.3 *5	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	5	10	15
RES.	20	25	4	9	14	19	24	3	8	13	18	23	2	7	12	17	22	1	6	11	16	21	26	5	10	15
P.4	T	Y	D	I	N	S	X	C	H	M	R	W	B	G	L	Q	V	A	F	K	P	U	Z	E	J	O

En el paso 2 hemos sumado 3 a las transformaciones del paso 1, y en el paso 3, los resultados obtenidos del paso anterior los hemos multiplicado por 5.

El alfabeto resultante es el siguiente:

ALFABETO: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A.CIFRADO: T Y D I N S X C H M R W B G L Q V A F K P U Z E J O

Así, dado el mensaje:

	N	O	S	E	R	A	E	L	L	U	N	E	S
PASO 1	14	15	19	5	18	1	5	12	12	21	14	5	19
PASO 2 +3	17	18	22	8	21	4	8	15	15	24	17	8	22
PASO 3 *5	85	90	110	40	105	20	40	75	75	120	85	40	110
RESIDUO	7	12	6	14	1	20	14	23	23	11	7	14	1
PASO 4	G	L	F	N	A	T	N	W	W	K	G	N	A

El paso 4 nos da el mensaje a enviar, que será:

GLFNATNWWKGNA

Otro método es el siguiente:

1. Se asigna a cada letra un valor numérico, por ejemplo, A=1, B=2, etcétera.

2. Establecemos una matriz, que posea inversa para poder después descifrar.

3. Se asigna a cada letra del mensaje su valor; sea, por ejemplo:

MENSAJE: SERA EL MARTES

que nos produce: 19051801 0512 130118200519; si el número de letras del mensaje hubiera sido impar, tendríamos que haber añadido una letra para que fuese par.

4. Se multiplican los números obtenidos en el paso anterior por la matriz preestablecida, sea ésta, por ejemplo:

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = \begin{bmatrix} 5 & 6 \\ 4 & 5 \end{bmatrix}$$

ahora obtendremos:

$$A = (x_{11}*y_1 + x_{12}*y_2) \text{ mód. } 26$$
$$B = (x_{21}*y_1 + x_{22}*y_2) \text{ mód. } 26$$

siendo y_1 e y_2, los números correspondientes a las letras 1.as y 2.as, así en nuestro ejemplo sería:

A = 19 y B = 05

después la 3.ª se vuelve a asociar a A y la 4.ª a B, y así sucesivamente:

A		B	
1.ª letra		2.ª letra	
3.ª	"	4.ª	"
5.ª	"	6.ª	"
7.ª	"	8.ª	"
9.ª	"	10.ª	"
11.ª	"	12.ª	"

de esta multiplicación, obtenemos:

21231825 1902 190502160911

lo que convertido a letras, nos da el mensaje a enviar:

TEXTO CIFRADO: UWRY SB SEBPIK

el método de descifrado es similar, sólo que hay que multiplicar por la matriz inversa de la establecida:

$$\begin{bmatrix} x_{22}/n & -x_{12}/n \\ -x_{21}/n & x_{11}/n \end{bmatrix}$$

siendo $n = x_{11}*x_{22} - x_{12}*x_{21}$

lo que en nuestro caso produce:

$$\begin{bmatrix} 5 & -6 \\ -4 & 5 \end{bmatrix}$$

Un programa que realiza este tipo de cifrado-descifrado es el siguiente:

```
10 REM METODO MATRIZ
17 LET A=64
20 CLS:REM <-- EN EL COMMODORE CAMBIARLO POR: PRIN
T "[SHIFT-HOME]"
30 DATA 1,9,21,15,3,19,0,7,23,11,5,17,25
40 LOCATE 1,1:PRINT "METODO MATRIZ PARA CIFRADO Y
DESCIFRADO"
45 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
50 LOCATE 3,1:INPUT "MODO (C/D) : ";N$
70 IF N$="F" THEN GOTO 680
80 IF N$<"C" OR N$>"D" THEN GOTO 50
110 LOCATE 3,1:PRINT "   MODO = ";N$;"       "
115 REM LEE LA MATRIZ
120 LOCATE 5,1:INPUT "MATRIZ (1,2,3,4) :";A1,A2,A3
,A4
130 REM CALCULA EL DETERMINANTE
160 LET DT=A1*A4-A2*A3
170 GOSUB 1000
180 LET TM=DT
190 IF DT=0 THEN LOCATE 5,1:PRINT "NO TIENE MATRIZ
 INVERSA        ":GOTO 120
230 IF (DT/2<= INT (DT/2)) OR (DT/13 >= INT (DT/13
)) THEN LOCATE 5,1:PRINT "EL DETERMINANTE NO ES VA
LIDO      ":GOTO 120
260 LOCATE 5,1:PRINT :MATRIZ(1,2,3,4) = ";A1;",";A
2;",";A3;",";A4
270 LOCATE 7,1:INPUT "MENSAJE = ";M$
310 REM LIMPIA LOS ESPACIOS DEL MENSAJE
380 GOSUB 5000
430 REM PREPARACION DEL MENSAJE
450 GOSUB 6000
460 LOCATE 7,1:PRINT "MENSAJE = ";M$;"
 "
480 IF N$="C" THEN GOSUB 2000:GOTO 500
490 GOSUB 3000
500 REM
640 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N) :
  "
650 LET R$=INKEY$:IF R$="" THEN GOTO 650:REM <-- E
N EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" THEN
 GOTO 650
```

```
660 IF R$<>"N" AND R$<>"S" THEN GOTO 650
670 IF R$="S" THEN RUN
680 REM FIN
690 END:REM <-- EN EL SPECTRUM CAMBIARLO POR: STOP
1000 REM AJUSTA EL RANGO DE LAS LETRAS DEL ALFABET
O
1010 IF DT<0 THEN LET DT=DT+26
1030 IF DT>=26 THEN LET DT=DT-26
1040 IF DT<0 OR DT>26 THEN GOTO 1010
1050 IF DT=0 THEN LET D1=26
1060 RETURN
2000 REM CIFRADO
2005 PRINT "MENSAJE CIFRADO = ";
2010 GOSUB 4000
2020 RETURN
3000 REM DESCIFRADO
3005 PRINT "MENSAJE DESCIFRADO = ";
3010 FOR N=1 TO T25 STEP 2
3020 READ M
3030 IF N=TM THEN LET R=M
3040 NEXT N
3050 LET N=A1:LET A1=A4:LET A4=N:LET A2=-A2:LET A3
=-A3
3100 LET DT=A1*R
3120 LET A1=DT
3130 LET DT=A2*R
3150 LET A2=DT
3160 LET DT=A3*R
3180 LET A3=DT
3190 LET DT=A4*R
3210 LET A4=DT
3230 GOSUB 4000
3240 RETURN
4000 REM IMPRIME MENSAJE
4020 FOR N=1 TO LM STEP 2
4030 LET D1=ASC(MID$(M$,N,1))-A:REM <-- EN EL SPEC
TRUM CAMBIARLO POR: LET D1=COD(M$(N))-A
4040 LET D2=ASC(MID$(M$,N+1,1))-A:REM <-- EN EL SP
ECTRUM CAMBIARLO POR: LET D2=COD(M$(N+1))-A
4050 LET DT=A1*D1+A2*D2
4060 GOSUB 100
4070 RETURN
4080 LET DT=A3*D1+4*D2
4090 GOSUB 1000
4100 PRINT CHR$(DT+A);
4110 NEXT N
4120 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SEAN L
ETRAS DEL MENSAJE
```

```
5010 LET I=2
5030 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET I=I-
1
5040 LET I=I+1
5050 IF I<=LEN(M$) THEN GOTO 5030
5060 LET LM=LEN(M$)
6000 REM RELLENA EL MENSAJE CON "W"
6010 LET LM=LEN(M$)
6020 IF LM/2<>INT(LM/2) THEN LET M$=M$+"W":LET LM=
LEN(M$)
6050 RETURN
```

COMENTARIOS DEL PROGRAMA MATRIZ

Inicialización (20-30). Se definen las constantes y la tabla de datos que empleará la rutina descifrada.

Cabecera y mensaje de finalización (40-45). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (50-80). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Imprime modo (110). Imprime el modo para visualizar la elección anterior.

Lectura de la matriz (120). Pide los coeficientes de la matriz de 2 por 2 que se empleará en el método.

Cálculo del determinante (160).

Ajusta rango (1000-1060). Ajusta el valor del determinante al rango de las letras del alfabeto.

Comprueba matriz (190-260). Si el determinante es igual a cero, no tiene matriz inversa y vuelve a pedir la matriz. Si el valor del determinante no es múltiplo de 2 o de 13, la matriz no es válida para el método y vuelve a pedirla.

Lectura del mensaje (270). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Prepara mensaje (6000-6050). Rellena el mensaje con «W» hasta conseguir que la longitud del mensaje sea un número múltiplo de 2.

Selección en función del modo (480). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2020). En función a la transformación de la matriz imprime el mensaje.

Descifrado (3000-3150). Obtiene la matriz transpuesta y a continuación obtiene la matriz inversa. En función a la transformación de la matriz obtenida imprime el mensaje.

Finalización o iteración (640-690). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Un método denominado homofónico opera como sigue: se asignan a cada símbolo del alfabeto claro un conjunto de símbolos homofónicos, pero el número de asignaciones guarda relación con la frecuencia de aparición de cada letra el lenguaje que se use; así, para el castellano, y asignando enteros entre 00 y 99, podría ser:

A 09,34,58,70,75,77,79,86	Ñ 54
B 13,72	N 04,53,56,65,66
C 29,31,42,91,96	O 12,68,69,71,80,94
D 48,50,84,87,97	P 07,24,26,47,90
E 39,49,51,76,81,83,85,88	Q 11,14,16
F 06,25	R 05,27,44,46
G 07,99	S 00,57,59,61,78,98
H 10,15	T 22,33,35,74,82,93
I 03,36,38,73,89	U 20,28,30,41,43,95
J 37	V 45
K 60	X 19
L 08,23,52,55,62,64	Y 01,02,18
M 21,32,40,63	Z 17

Así, las letras del mensaje «GUERRA», podrían tener cualquier conjunto de valores, que posean las letras que lo forman,

G: 67,99
U: 20,28,30,41,43,95
E: 39,49,51,76,81,83,85,88
R: 05,27,44,46
R: 05,27,44,46
A: 09,34,58,70,75,77,79,86

Un método desarrollado por Jefferson Beale, y que lleva su nombre, consiste en que, dado un mensaje claro, se le asigna un entero que corresponde a la situación de cada letra en un texto prefijado; sea éste el siguiente:

«Los ascendientes no siempre la nobleza otorgan, y, ¿que es la vida junto a un niño?; hijo, es felicidad.»

Y el mensaje a enviar sea: NO VENGAS (los espacios entre palabras en el texto prefijado no se cuentan), sería enviado así:

0802490708380405

no obligatoriamente tenemos que asignar a cada letra del mensaje que queremos criptografiar el valor numérico de la primera vez que encontramos en el texto esa letra; esto producirá una mayor dificultad a la hora de un criptoanálisis; así, la letra E puede tomar los valores: 7, 11, 14, etc.

Los tipos de sistemas de sustitución son:



Monoalfabéticos

Cuando se utiliza un solo alfabeto cifrado, como los de los ejemplos anteriores.

Otro procedimiento de obtener el alfabeto cifrado es a partir de una palabra clave, y consta de cinco fases:

1. Elección de la palabra clave, por ejemplo, CRIPTOGRAFIA.
2. Eliminación de caracteres repetidos en la palabra clave, en este caso quedaría: CRIPTOGAF.
3. Inserción de las letras restantes del alfabeto; para nuestro ejemplo sería: CRIPTOGAFBDEHJKLMNQSUVWXYZ.
4. Situar las letras en n columnas; supogamos que n=3

C R I P T O G A F
B D E H J K L M N
Q S U V W X Y Z

5. Colocar las letras continuadamente, pero enpezando por la columna 1 hasta la columna n; en nuestro caso n=9, y para el ejemplo propuesto, quedaría un alfabeto de la forma:

C B Q R D S I E U P H V T J W O K X G L Y A M Z F N

así, dado el mensaje: NO HAY PASO
enviaríamos: JPECFOCGP

Una variante a este método sería leer las columnas de abajo arriba, con lo que para nuestro ejemplo obtendríamos el alfabeto siguiente:

Q B C S D R U E I V H P W J T X K O Y L G Z M A N F

y nuestro mensaje sería enviado como:

JTEQNXQYT

Este método consigue un alfabeto mejor mezclado y, por tanto, más difícil de descifrar.

Un programa que nos cifre o descifre estos últimos métodos expuestos es el siguiente:

```
10 REM METODO MONOALFABETICO DE CIFRADO Y DESCIFRA
DO
12 LET C$=""
13 LET A=64
30 CLS:REM <-- EN EL COMMODORE CAMBIARLO POR: PRIN
T "[SHIFT-HOME]";
40 LOCATE 1,1: PRINT "METODO MONOALFABETICO DE CIF
RADO Y DESCIFRADO"
50 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
60 LOCATE 3,1:INPUT "MODO (C/D) : ";N$
80 IF N$="F" THEN GOTO 760
82 IF N$<"C" OR N$>"D" THEN GOTO 60
85 LOCATE 5,1:INPUT "CLASE (T/M) : ";L$
130 IF L$<>"T" AND L$<>"M" THEN GOTO 85
135 LOCATE 7,1:INPUT "CLAVE : ";C$
137 LET LC=LEN(C$)
138 IF LC<1 OR LC>26 THEN GOTO 135
139 LET O$="N":LET I=1
140 IF MID$(C$,I,1)<"A" OR MID$(C$,I,1)>"Z" THEN L
ET O$="S"
155 LET I=I+1
160 IF I<LC AND O$<>"S" THEN GOTO 140
170 IF O$="S" THEN GOTO 135
180 LOCATE 3,1:PRINT "        MODO = ";N$;"        "
190 LOCATE 5,1:PRINT "       CLASE = ";L$;"        "
200 LOCATE 7,1:PRINT "       CLAVE = ";C$;"        "
210 REM PREPARACION DE LA CLAVE
220 LET C$=C$+"ABCDEFGHIJKLMNOPQRSTUVWXYZ"
230 LET L1=LEN(C$)
240 GOSUB 1000
650 LOCATE 9,1:INPUT "     MENSAJE = ";M$
692 REM ELIMINA LOS ESPACIOS DEL MENSAJE
695 GOSUB 5000
700 IF N$="C" THEN GOSUB 2000:GOTO 710
705 GOSUB 3000
710 REM
720 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N) :
  "
730 LET R$=INKEY$:IF R$="" THEN GOTO 730:REM <-- P
ARA EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" TH
EN GOTO 730
740 IF R$<>"S" AND R$<>"N" THEN GOTO 730
750 IF R$="S" THEN RUN
```

```
760 REM FIN
770 END:REM <-- PARA EL SPECTRUM CAMBIARLO POR: ST
OP
1000 REM PREPARACION DE LA CLAVE
1040 DIM A$(27):FOR I=1 TO 27:LET A$(I)=" ":NEXT I
1050 FOR I=1 TO L1
1060 LET AS=ASC(MID$(C$,I,1))-A:REM <-- EN EL SPEC
TRUM CAMBIARLO POR: LET AS=COD(C$(I))-A
1070 IF A$(AS+1)=" " THEN LET T$=T$+MID$(C$,I,1):R
EM <-- EN EL SPECTRUM CAMBIAR LA ULTIMA PARTE POR:
 LET T$=T$+C$(I)
1090 IF I=LC THEN LET Y=LEN(T$)
1110 LET A$(AS+1)="&"
1120 NEXT I
1130 IF L$="M" THEN RETURN
1135 REM TRANSPOSICION
1140 LET X=INT(26/Y)
1150 IF Y*X<>26 THEN LET X=X+1:GOTO 1180
1160 LET TM=Y*X
1180 IF TM>26 THEN FOR I=27 TO TM:LET T$=T$+"&":NE
XT I
1220 ERASE A$:DIM A$(X,Y):REM <-- EN EL SPECTRUM C
AMBIARLO POR: LET A$="":DIM A$(X,Y)
1240 LET K=1
1250 FOR J=1 TO X
1260 FOR I=1 TO Y
1270 LET A$(J,I)=MID$(T$,K,1):REM <-- EN EL SPECTR
UM CAMBIARLO POR: LET A$(J,I)=T$(K)
1280 LET K=K+1
1290 NEXT I
1300 NEXT J
1310 LET T$=""
1320 FOR I=1 TO Y
1330 FOR J=X TO 1 STEP -1
1340 IF A$(J,I)<>"&" THEN LET T$=T$+A$(J,I)
1360 NEXT J
1370 NEXT I
1375 ERASE A$:REM <-- EN EL SPECTRUM CAMBIARLO POR
 A$=""
1380 RETURN
2000 REM CIFRADO
2010 PRINT "MENSAJE CIFRADO = ";
2020 GOSUB 4000
2030 RETURN
3000 REM DESCIFRADO
3010 DIM P$(26)
3015 PRINT "MENSAJE DESCIFRADO = ";
3020 FOR I=1 TO 26
```

```
3030 LET PP=ASC(MID$(T$,I,1))-A:REM <-- EN EL SPEC
TRUM CAMBIARLO POR: LET PP=CODE(T$(I))-A
3040 LET P$(PP)=CHR$(A+I)
3050 NEXT I
3060 FOR I=1 TO 26
3070 MID$(T$,I,1)=P$(I):REM <-- EN EL SPECTRUM CAM
BIARLO POR: LET T$(I)=P$(I)
3080 NEXT I
3090 GOSUB 4000
3100 RETURN
4000 REM IMPRIME EL MENSAJE
4010 FOR I=1 TO LM
4020 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
PRINT " ";:GOTO 4060
4030 PRINT MID$(T$,ASC(MID$(M$,I,1))-A,1);
4060 NEXT I
4070 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SEAN L
ETRAS DEL MENSAJE
5010 LET I=2
5030 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET I=I-
1
5040 LET I=I+1
5050 IF I<LEN(M$) THEN GOTO 5030
5060 LET LM=LEN(M$)
5070 RETURN
```

COMENTARIOS DEL PROGRAMA MONOALFABETICO

Inicialización (10-13). Se definen las constantes y se inicializan algunas variables que empleará el programa.

Cabecera y mensaje de finalización (40-50). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (60-82). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clase (85-130). Pide la clase de cifrado que se empleará en el método; éste puede ser «T» para transposición y «M» para mezcla. Si la tecla pulsada no es ninguna de las anteriores, el programa vuelve a pedir la clase.

Lectura de la clave (135-170). Pide la clave del método. Esta clave debe ser una cadena de caracteres. Si la clave introducida tiene caracteres no

válidos, que no pertenecen al alfabeto (A..Z), el programa vuelve a pedir la clave.

Preparación de la clave (1000-1380). Obtiene la transformación de la clave en función a la clase. Si se trata de transposición, el alfabeto que tenemos se transpone al meterlo y volver a sacarlo de la matriz.

Imprime modo, clase y clave (180-200). Imprime el modo, la clase y la clave para visualizar la elección anterior.

Lectura del mensaje (650). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Selección en función del modo (700). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2030). Para cada uno de los caracteres del mensaje se le aplica la transformación, obteniendo la letra que ocupa la posición correspondiente del mensaje.

Descifrado (3000-3110). Para cada uno de los caracteres del mensaje se le aplica la transformación inversa, ordenando de forma que cada elemento sea equivalente a la letra que ocupa la posición correspondiente del mensaje cifrado.

Finalización o iteración (720-770). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Para resolver los criptogramas de sustitución monoalfabéticos, y dado que éstos siempre muestran la estructura de las palabras del texto original, usaremos la relación característica entre letras y su frecuencia de aparición, que muestran todos los lenguajes; a continuación se dan las frecuencias de aparición de letras, frecuencia de aparición de letras iniciales, frecuencia de aparición de letras finales y diagrama o frecuencia de aparición de dos letras juntas, para los idiomas castellano e inglés.

CASTELLANO

Frecuencia de letras ordenadas alfabéticamente y por frecuencias (%).				Frecuencia de aparición de letras iniciales en palabras, ordenadas alfabéticamente y por frecuencias (%).				Frecuencia de aparición de letras finales en palabras, ordenadas alfabéticamente y por frecuencias (%).			
A	12,0	E	16,8	A	7,1	D	17,3	A	20,1	E	24,1
B	0,9	A	12,0	B	0,4	E	14,8	B	0,0	A	20,1
C	2,9	O	8,7	C	4,6	L	13,6	C	0,2	S	13,0
D	6,9	L	8,4	D	17,3	P	7,2	D	0,8	N	11,4

E	16,8	S	7,9	E	14,8	A	7,1	E	24,1	L	10,5
F	0,5	N	7,0	F	1,2	S	7,0	F	0,0	O	10,0
G	0,7	D	6,9	G	0,8	Q	4,8	G	0,0	Y	4,1
H	0,8	R	5,0	H	2,5	C	4,6	H	0,0	R	3,7
I	4,1	U	4,8	I	0,4	Y	3,9	I	0,7	U	1,0
J	0,3	I	4,1	J	0,5	U	3,4	J	0,0	D	0,8
K	0,1	T	3,3	K	0,1	M	3,3	K	0,0	I	0,7
L	8,4	C	2,9	L	13,6	H	2,5	L	10,5	Z	0,3
M	2,1	P	2,7	M	3,3	T	2,4	M	0,0	C	0,2
N	7,0	M	2,1	N	2,2	N	2,3	N	11,4	T	0,1
Ñ	0,3	Y	1,5	Ñ	0,0	F	1,2	Ñ	0,0	B	0,0
O	8,7	Q	1,5	O	1,1	O	1,1	O	10,0	F	0,0
P	2,7	B	0,9	P	7,2	G	0,8	P	0,0	G	0,0
Q	1,5	H	0,8	Q	4,8	R	0,7	Q	0,0	H	0,0
R	5,0	G	0,7	R	0,7	J	0,5	R	3,7	J	0,0
S	7,9	F	0,5	S	7,0	I	0,4	S	13,0	K	0,0
T	3,3	V	0,4	T	2,4	B	0,4	T	0,1	M	0,0
U	4,8	Ñ	0,3	U	3,4	V	0,3	U	1,0	Ñ	0,0
V	0,4	J	0,3	V	0,3	Z	0,2	V	0,0	P	0,0
X	0,2	Z	0,2	X	0,1	K	0,1	X	0,0	Q	0,0
Y	1,5	X	0,2	Y	3,9	X	0,1	Y	4,1	V	0,0
Z	0,2	K	0,1	Z	0,2	Ñ	0,0	Z	0,3	X	0,0

Tabla de digramas

INGLES

A	8,0	E	13,0	A	11,1	T	16,5	A	3,0	E	20,0
B	1,5	T	9,0	B	4,7	A	11,1	B	0,3	S	12,5
C	3,0	A	8,0	C	5,8	S	7,5	C	0,6	D	10,0
D	4,0	O	8,0	D	2,9	O	5,8	D	10,0	T	9,6
E	13,0	N	7,0	E	2,6	C	5,7	E	20,0	N	9,5
F	2,0	R	6,5	F	4,1	I	5,1	F	4,6	Y	5,5
G	1,5	I	6,5	G	1,9	W	4,8	G	2,8	R	5,4
H	6,0	H	6,0	H	3,9	P	4,7	H	2,5	F	4,6
I	6,5	S	6,0	I	5,7	B	4,1	I	0,5	O	4,5
J	0,5	D	4,0	J	0,7	F	3,9	J	0,2	L	3,7
K	0,5	L	3,5	K	0,6	H	3,6	K	1,0	A	3,0
L	3,5	U	3,0	L	2,2	M	3,2	L	3,7	G	2,8
M	3,0	C	3,0	M	3,6	R	2,9	M	1,3	H	2,5
N	7,0	M	3,0	N	2,5	D	2,6	N	9,5	M	1,3
O	8,0	F	2,0	O	7,2	E	2,5	O	4,5	K	1,0
P	2,0	P	2,0	P	4,8	N	2,2	P	0,5	W	1,0
Q	0,2	Y	2,0	Q	0,3	L	1,9	Q	0,1	C	0,6
R	6,5	B	1,5	R	3,2	G	1,4	R	5,4	I	0,5

S	6,0	G	1,5	S	7,5	U	0,8	S	12,5	P	0,5
T	9,0	W	1,5	T	16,5	Y	0,7	T	9,6	B	0,3
U	3,0	V	1,0	U	1,4	J	0,7	U	0,3	U	0,3
V	1,0	J	0,5	V	0,7	V	0,6	V	0,2	X	0,3
W	1,5	K	0,5	W	5,1	K	0,3	W	1,0	V	0,2
X	0,5	X	0,5	X	0,1	Q	0,1	X	0,3	J	0,2
Y	2,0	Z	0,2	Y	0,8	X	0,1	Y	5,5	Q	0,1
Z	0,2	Q	0,2	Z	0,1	Z		Z	0,1	Z	0,1

Tabla de digramas

Cuadro de diagramas (frecuencia de aparición de dos letras juntas) en castellano

1.ª letra \ 2.ª letra	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A		–	+	·		×	×	×	×	×		·	–	·		×		·	$	×	×	×	+		–	×
B	–				–				–			–			×			–			×					
C	+		×		–			×	·			×		×	·			×		–	–					
D	+				$				–				×		·			–			×					
E	×	×	+	–	×	×	–		×	×		$	+	$	×	×	×	$	$	×	×	–	×	×	×	×
F	×				–				×			×			×			×			–					
G	–				×				×						–			×		×	–					
H	·				×				×						–						×					
I	+	×	+	+	+		×		×	×		–	–	+	·	×		×	+	–	×	×				×
J	×														–						–					
K									×																	
L	$				–		×		–			–	×		$		×			×	×					
M	·	–			+				+						+	–					–					
N	·		–	–	+	×	×		+	×				·			×		×	·	–	×				×
Ñ	×								×					×	–											
O		–	–	–	×	×	×					+	+	$		×		$	$	–				×	×	×
P	·				–				×			×		×	·			+	×	×	+					
Q																					$					
R	$		×	×	·		×		+			×	×	–	·		×	×	×	–	×	×				×
S	–		×	×	$				·				×		+	–				·	+					
T	·	×		·	×				+						·			·			–	×				
U	–	×	×	×	$		×		×			–	×	$		×		–	–	×		×			×	
V	–				×				–						×						×					
X									×											×						
Y	×				–										×						×					
Z	×														×											

$ Frecuencia de aparición muy alta. · frecuencia de aparición alta. + Frecuencia de aparición media. – Frecuencia de aparición baja. × Frecuencia de aparición rara. Blanco: no aparece.

Cuadro de diagramas (frecuencia de aparición de dos letras juntas) en inglés

1.ª letra \ 2.ª letra	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	×	−	−	+	×	×	−	×	+	×	×	·	−	$	×	−		·	·	×	−	×	×	−	×	
B	−	×	×	×	+				×	×		−	×		−			×	×	×	×	×			−	
C	+		×	×	+			+	−		×	×	×	×	·		×	−	×	−	×				×	
D	−	×	×	×	·	×	×	×	−	×		×	×	×	−		×	×	×	×	×	×	×		×	
E	+	×	+	·	+	−	×	×	−	×	×	+	−	$	×	−	×	$	·	+	×	−	−	−	×	×
F	×				−	−			−			×	×	×	+			−	×	×	×				×	
G	−			×	−		×	−	×			×	×	×	−			−	×	×	×				×	
H	·	×	×	×	$			×	+			×	×	×	−			×	×	×	×		×		×	
I	−	×	+	−	−	×	−	×	×		×	+	−	$	·	×	×	−	·	·	×	−		×	×	×
J	×				×				×						×			×			×					
K	×	×		×	−	×		×	×		×	×	×	×	×			×	×	×	×				×	
L	+	×	×	−	·	×	×		+		×	+	×	×	−	×		×	−	×	×	×	×	×	−	
M	+	×	×		·	×			−			×	−	×	−	−		×	×	×	×	×			×	
N	−	×	−	·	·	×	·	×	+	×	×	×	×	×	−	×	×	×	+	·	×	×	×	×	×	×
O	×	×	×	−	×	·	×	×	×	×	×	−	+	$	−	−		·	−	−	·	−	−	×	×	×
P	−		×		+			×	×			−	×	×	−	−		+	×	×	×				×	
Q																					×					
R	+	×	×	−	$	×	×	×	·		×	×	−	−	·	×		×	−	−	×	×	×	×	−	
S	−	×	×	×	·	×		−	+		×	×	×	×	−	−	×	×	−	·	−		×		×	
T	+	×	×	×	·	×	×	$	·			×	×	×	·	×		−	−	−	−	×	×		−	×
U	×	×	−	×	×	×	×	×	×		×	−	×	+	×	×		+	−	−	×	×		×	×	×
V	×			×	·				−				×		×				×		×	×			×	
W	−	×		×	−			−	−			×		×	−			×	×	×					×	
X	×		×		×			×	×				×		×	×				×				×		
Y	×	×	×	×	×				×		×	×	×	×	×	×		×	×	×	×		×		×	×
Z	×				×				×			×			×						×					×

$ Frecuencia de aparición muy alta. · frecuencia de aparición alta. + Frecuencia de aparición media. – Frecuencia de aparición baja. × Frecuencia de aparición rara. Blanco: no aparece.

Con estas tablas de frecuencia se realiza un detallado análisis del texto cifrado para tratar de encontrar el mensaje claro; supongamos que interceptamos el siguiente mensaje, que sabemos está en castellano:

TEQC EJ KCQSER

vemos que la T aparece 1 vez;
vemos que la J aparece 1 vez;
vemos que la K aparece 1 vez;
vemos que la S aparece 1 vez;
vemos que la R aparece 1 vez;
vemos que la Q aparece 2 veces;
vemos que la C aparece 2 veces;
vemos que la E aparece 3 veces.

En castellano la letra de mayor frecuencia es la E, la asociamos a la E del mensaje.

Después la Q y la C aparecen dos veces; la letra de mayor frecuencia de aparición después de la E es la A, o sea, que teóricamente o la Q o la C deben ser la A; supongamos que la C la sustituimos por la A, dado que así será el final de la primera palabra, lo que es muy frecuente, con lo que ya tenemos:

```
TEQC EJKCQSER
_E_A E__A___E_
```

la J, que sigue a la E según la tabla de diagramas puede ser una L, N, R o S; la R, obviamente, no puede ser, pues en castellano no existe ER como palabra, pero sí EL, EN o ES, pero si nuestra tabla de diagramas fuera más exacta, en cifras, veríamos que la mayor frecuencia se produce para EL, por lo que la L la asociamos a la J.

Estudiaremos ahora la Q; si nos fijamos en la primera palabra vemos que va seguida de la A y según nuestra tabla de diagramas las mayores posibilidades de anteceder a la A son para la L o la R, con lo que tendríamos:

```
TEQC EJKCQSER
_ERA EL_AR_ES
  L      L
```

En la primera palabra, y según nuestra tabla de diagramas, la E tiene como mayores probabilidades de llevar delante la D, S y la U, solamente la S forma una palabra lógica: SERA, por lo que acomodamos la S a la T del texto cifrado, con lo que la Q del texto cifrado la acomodamos definitivamente a la R y obtenemos:

```
TEQC EJKCQSER
SERA EL_AR_ES
```

En este punto ya es fácil descubrir las dos letras ocultas, que son la M y la T, obteniendo el mensaje claro:

SERA EL MARTES

El siguiente programa realiza un análisis de frecuencia para:

a) Aparición de cada letra;
b) Tabla de diagramas;
c) Aparición de letras iniciales;
d) Aparición de letras finales.

gracias a él nos evitamos el tener que obtener a mano todos estos datos.

Un programa que realiza estas estadísticas de frecuencias es el siguiente:

```
10 REM METODO DE FRECUENCIAS
15 LET A=64
20 DIM B(26):DIM C(26):DIM E(26):DIM D(26):DIM T(2
6,26):DIM F(26,26)
30 CLS:REM <-- PARA EL COMMODORE CAMBIARLO POR: PR
INT "[SHIFT-HOME]"
40 LOCATE 1,1:PRINT "METODO FRECUENCIAS DE CIFRADO
 Y DESCIFRADO"
50 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
70 LOCATE 3,1:INPUT "MENSAJE : ";M$:LET Z$=M$
80 IF LEFT$(M$,1)="F" THEN GOTO 560:REM <-- EN EL
SPECTRUM CAMBIARLO POR: IF M$(1)="F" THEN GOTO 560
100 REM LIMPIA EL MENSAJE
110 GOSUB 5000
120 REM LIMPIA EL MENSAJE MENOS LOS ESPACIOS
130 GOSUB 6000
190 LOCATE 3,1:PRINT "MENSAJE = ";Z$
300 GOSUB 2000:REM FRECUENCIA RELATIVA DE LAS LETR
 AS
 310 GOSUB 3000:REM FRECUENCIA DE DOS LETRAS CONSEC
 UTIVAS
 320 GOSUB 4000:REM FRECUENCIA DE LETRAS INICIALES
 Y FINALES
 510 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N) :
  "
 520 LET R$=INKEY$:IF R$="" THEN GOTO 520:REM <-- E
 N EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" THEN
  GOTO 520
 530 IF R$<>"S" AND R$<>"N" THEN GOTO 520
 540 IF R$="S" THEN RUN
 550 REM FIN
 560 END:REM <-- EN EL SPECTRUM CAMBIARLO POR: STOP
 2000 REM CALCULO DE FRECUENCIAS
 2010 LOCATE 5,1:PRINT "FRECUENCIA RELATIVA DE LAS
 LETRAS"
 2020 FOR I=1 TO LM
 2025 LET NL=ASC(MID$(M$,I,1)):REM <-- EN EL SPECTR
 UM CAMBIARLO POR: LET NL=CODE(M$(I))
 2030 LET E(NL-A)=E(NL-A)+1
 2040 NEXT I
 2050 FOR I=1 TO 26
 2070 PRINT CHR$(I+A);"=";E(I)/LM*100;"  ";
 2080 NEXT I
 2090 RETURN
 3000 REM
```

```
3010 LOCATE 11,1:PRINT "FRECUENCIA DE DOS LETRAS C
ONSECUTIVAS"
3020 FOR I=1 TO LM-1
3025 LET L1=ASC(MID$(M$,I,1)):LET L2=ASC(MID$(M$,I
+1,1)):REM <-- EN EL SPECTRUM CAMBIARLO POR: LET L
1=CODE(M$(I)):LET L2=CODE(M$(I+1))
3030 LET T(L1-A,L2-A)=T(L1-A,L2-A)+1
3040 NEXT I
3050 FOR I=1 TO 26
3060 FOR J=1 TO 26
3070 IF T(I,J)>0 THEN PRINT CHR$(I+A);CHR$(J+A);"=
";T(I,J);"  ";:LET CC=CC+1
3080 NEXT J
3090 NEXT I
3100 RETURN
4000 REM
4010 FOR I=1 TO LEN(Z$)
4015 LET X$=MID$(Z$,I,1):LET Y$=MID$(Z$,I+1,1):REM
 <-- EN EL SPECTRUM CAMBIARLO POR: LET X$=Z$(I):LE
T Y$=Z$(I+1)
4020 IF I=1 THEN LET B(ASC(X$)-A)=B(ASC(X$)-A)+1:R
EM <-- EN EL SPECTRUM DONDE PONE "ASC" CAMBIARLO P
OR "CODE"
4030 IF I=LEN(Z$) THEN LET C(ASC(X$)-A)=C(ASC(X$)-
A)+1
4040 IF X$<>" " AND Y$=" " THEN LET C(ASC(X$)-A)=C
(ASC(X$)-A)+1
4050 IF X$=" " AND Y$<>" " THEN LET B(ASC(Y$)-A)=B
(ASC(Y$)-A)+1
4060 NEXT I
4070 REM
4080 PRINT "LETRAS INICIALES"
4090 FOR I=1 TO 26
4100 IF B(I)<>0 THEN PRINT CHR$(I+A);"=";B(I);"   "
;
4110 NEXT I
4120 PRINT "LETRAS FINALES"
4130 FOR I=1 TO 26
4140 IF C(I)<>0 THEN PRINT CHR$(I+A);"=";C(I);"   "
;
4150 NEXT I
4160 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SEAN L
ETRAS DEL MENSAJE
5010 LET I=2
5030 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET I=I-
1
```

```
5040 LET I=I+1
5050 IF I<LEN(M$) THEN GOTO 5030
5060 LET LM=LEN(M$)
5070 RETURN
6000 REM LIMPIA TODOS LOS CARACTERS QUE NO SEAN LE
TRAS DEL MENSAJE
6010 LET I=2
6030 IF (L$<"A" OR L$>"Z") AND L$<>" " THEN LET Z$
=LEFT$(Z$,I-1)+RIGHT$(Z$,LEN(Z$)-I):LET I=I-1
6040 LET I=I+1
6050 IF I<=LEN(Z$) THEN GOTO 6030
6060 RETURN
```

COMENTARIOS DEL PROGRAMA FRECUENCIAS

Inicialización (10-20). Se definen las constantes y las tablas que empleará el programa.

Cabecera y mensaje de finalización (40-60). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea, indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del mensaje (70). Pide el mensaje para obtener diferentes estadísticas de frecuencia.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Frecuencia relativa de las letras (2000-2090). Calcula la frecuencia relativa de cada una de las letras.

Frecuencia de dos letras consecutivas (3000-3100). Calcula la frecuencia de grupos de dos letras consecutivas.

Frecuencia de letras iniciales y finales (4000-4160). Calcula la frecuencia de las letras iniciales y finales.

Finalización o iteración (510-570). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Se podría realizar un programa para sugerir las letras más probables, en ciertos puntos del criptograma, pero necesitaríamos usar potentes métodos estadísticos y un computador con gran memoria, por lo que no se describirá aquí.

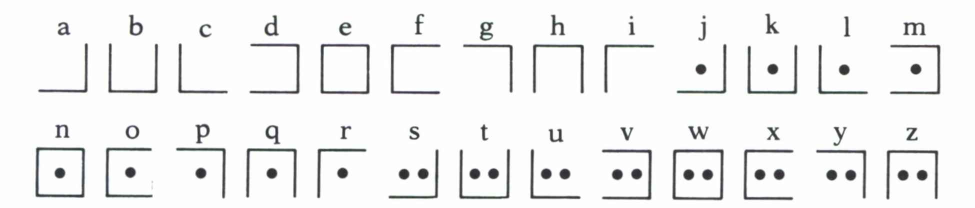
Un curioso alfabeto simbólico, usado por Edgar Allen Poe en su novela *El escarabajo de oro,* es el siguiente, en el que se asocia el alfabeto a tres matrices de 3 por 3:

a	b	c
d	e	f
g	h	i

j	k	l
m	n	o
p	q	r

s	t	u
v	w	x
y	z	

donde la correspondencia con el albabeto simbólico es la siguiente:



Así, la palabra CONFIDENCIAL se transmitiría como:



También se puede expresar este alfabeto, en números, de la forma siguiente:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	.1	.2	.3	$\underset{\cdot}{1}$	$\underset{\cdot}{2}$	$\underset{\cdot}{3}$	$\dot{1}$	$\dot{2}$	$\dot{3}$	1.	2.	3.	:1	:2	:3	$\ddot{1}$	$\ddot{2}$	$\ddot{3}$	1:	2:	3:	$\underset{\cdot\cdot}{1}$	$\underset{\cdot\cdot}{2}$

Con lo que nuestro mensaje CONFIDENCIAL sería enviado como:

3 3. 2. .3 $\underset{\cdot}{3}$ .1 .2 2. 3 $\underset{\cdot}{3}$ 1 $\dot{3}$



Polialfabéticos

Se utilizan múltiples alfabetos cifrados, como la Tabla del cifrado «Vigenere», que se da en la pág. 58, y que opera como sigue:

1. Se escoge una clave, que puede ser una palabra o frase de igual longitud al mensaje a criptografiar; si usamos una palabra y es de menor longuitud del mensaje a cifrar, repetimos la palabra tantas veces como sea necesario, hasta obtener igual longitud que el mensaje; la situamos encima del texto claro; así, por ejemplo:

CLAVE: CRIPTOGRAFIACRIP

TEXTO: ESTEESELMENSAJEX

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

2. Cada carácter es cifrado mediante la tabla Vigenere, donde la fila representa la clave y la columna el texto. Así, el primer carácter de la clave C y el primero del texto E dan G, y así sucesivamente, resultando:

TEXTO CIFRADO: GJBTXGKCMJVSCAMM

A partir de este método se pueden incorporar variaciones para ocultar aún más la información.

Un programa que realiza el cifrado-descifrado de este tipo es el siguiente:

```
10 REM METODO POLIALFABETICO DE CIFRADO Y DESCIFRADO
15 LET BL=32
16 LET A=65
```

```
17 LET Z=90
18 LET C=0
20 CLS:REM <-- EN EL COMMODORE CAMBIARLO POR: PRIN
T "[SHIFT-HOME]"
30 LOCATE 1,1:PRINT "METODO POLIALFABETICO DE CIFR
ADO Y DESCIFRADO"
40 LOCATE 20,1:PRINT "PARA FINALIZAR PULSE F"
50 LOCATE 3,1:INPUT "MODO (C/D) : ";N$
60 IF N$="F" THEN GOTO 540
70 IF N$<"A" OR N$>"Z" THEN GOTO 50
80 LOCATE 5,1:INPUT "CLAVE : ";C$
90 LET LC=LEN(C$)
100 IF LC<1 OR LC>26 THEN GOTO 80
110 REM COMPROBACION Y ALMACENAMIENTO DE LA CLAVE
120 DIM C(LC-1)
130 GOSUB 1000
155 IF O$="S" THEN RUN
160 LOCATE 3,1:PRINT "      MODO = ";N$
170 LOCATE 5,1:PRINT "     CLAVE = ";C$
180 LOCATE 7,1:INPUT "  MENSAJE = ";M$
200 REM LIMPIA ESPACIOS DEL MENSAJE
210 GOSUB 5000
230 REM
280 IF N$="C" THEN GOSUB 2000:GOTO 500
290 GOSUB 3000
500 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N) :
  "
510 LET R$=INKEY$:IF R$="" THEN GOTO 510:REM <-- E
N EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" THEN
 GOTO 510
520 IF R$<>"S" AND R$<>"N" THEN GOTO 510
530 IF R$="S" THEN RUN
540 REM FIN
550 END:REM <-- EN EL SPECTRUM CAMBIARLO POR: STOP
1000 REM COMPRUEBA LA CLAVE Y LA ALMACENA
1010 LET O$="N":LET I=1
1020 LET D$=MID$(C$,I,1):REM <-- PARA EL SPECTRUM
CAMBIARLO POR: LET D$=C$(I)
1040 IF D$<"A" OR D$>"Z" THEN LET O$="S"
1050 LET C(I)=ASC(D$)-A:REM <-- PARA EL SPECTRUM C
AMBIARLO POR: LET C(I)=CODE(D$)-A
1060 LET I=I+1
1070 IF I<=LC AND O$<>"S" THEN GOTO 1020
1080 RETURN
2000 REM CIFRADO
2010 PRINT "MENSAJE CIFRADO : ";
2020 FOR I=1 TO LM
2040 LET C=C+1
```

```
2050 IF C=LC+1 THEN LET C=1
2060 LET L=ASC(MID$(M$,I,1))+C(C):REM <-- EN EL SP
ECTRUM CAMBIARLO POR: LET L=CODE(M$(I))+C(C)
2070 IF L>Z THEN LET L=L-26
2080 PRINT CHR$(L);
2090 NEXT I
2100 RETURN
3000 REM DESCIFRADO
3010 PRINT "MENSAJE DESCIFRADO : ";
3020 FOR I=1 TO LM
3040 LET C=C+1
3050 IF C=LC+1 THEN LET C=1
3060 LET L=ASC(MID$(M$,I,1))-C(C):REM <-- EN EL SP
ECTRUM CAMBIARLO POR: LET L=CODE(M$(I))-C(C)
3070 IF L1A THEN LET L=L+26
3080 NEXT I
3100 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SEAN L
ETRAS DEL MENSAJE
5010 LET I=2
5030 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET I=I-
1
5040 LET I=I+1
5050 IF I<LEN(M$) THEN GOTO 5030
5060 LET LM=LEN(M$)
5070 RETURN
```

COMENTARIOS DEL PROGRAMA POLIALFABETICO

Inicialización (10-18). Se definen las constantes y algunas de las variables que empleará el programa.

Cabecera y mensaje de finalización (20-40). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (50-70). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clave (80-155). Pide la clave del método. Esta clave debe ser una cadena de caracteres. Si la clave introducida tiene caracteres no válidos, que no pertenecen al alfabeto (A..Z), o si la longitud no está comprendida entre 1 y 26, el programa vuelve a pedir la clave.

Imprime modo y clave (160-170). Imprime el modo y la clave para visualizar la elección anterior.

Lectura del mensaje (180). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

Selección en función del modo (280). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2100). Para cada uno de los caracteres del mensaje se le aplica el desplazamiento (positivo) de la clave y a continuación se imprime.

Descifrado (3000-3100). Para cada uno de los caracteres del mensaje se le aplica el desaplazamiento (negativo) de la clave y a continuación se imprime.

Finalización o iteración (500-550). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

Otra variante de este tipo de cifrados es el siguiente:

Dada la tabla siguiente:

A B	A	B	C	D	E	F	G	H	I	J	K	L	M
C D	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
E F	A	B	C	D	E	F	G	H	I	J	K	L	M
G H	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
I J	A	B	C	D	E	F	G	H	I	J	K	L	M
K L	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Q R	A	B	C	D	E	F	G	H	I	J	K	L	M
S T	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M N	A	B	C	D	E	F	G	H	I	J	K	L	M
O P	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
U V	A	B	C	D	E	F	G	H	I	J	K	L	M
W X	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

1. Se escoge una clave, que puede ser una palabra o frase de igual longitud al mensaje a criptografiar; si usamos una palabra y es de menor longuitud que el mensaje a cifrar, repetimos la palabra tantas veces como sea necesario, hasta obtener igual longitud que el mensaje; la situamos encima del texto claro; así, por ejemplo:

LLAVE: CRIPTOGRAFIACRIP

TEXTO: ESTEESELMENSAJEX

2. Se busca en los cuadros de la izquierda la letra de la llave, eso nos da el alfabeto a utilizar, que es el de la derecha de donde está la llave. Si la letra está en la línea superior, se sustituye por la que está debajo; si está en la línea inferior, se sustituye por la que está encima.

Así, para nuestro ejemplo,

TEXTO CIFRADO: RFGRRFRYZRAFNWRK



Sustitución Digráfica

En lugar de procesar, carácter a carácter, la sustitución digráfica sustituye los caracteres de dos en dos. Así, el sistema «Playfair» utiliza una matriz basada en una llave, como, por ejemplo:

LLAVE:	O	R	D	E	N
RESTO	A	B	C	F	G
LETRAS	H	IJ	K	L	M
ALFABETO:	P	Q	S	T	U
	V	W	X	Y	Z

El proceso es como sigue:

1. Se divide el texto en grupos de dos caracteres; si los caracteres en algún grupo son iguales, se separan y al primero de ellos se le añade un carácter poco frecuente, como la w. Por ejemplo:

TEXTO: ESTO ES UN MENSAJE

PASO 1: ES TO ES UN ME NS AJ EW

2. Se van sustituyendo cada par de caracteres, de forma que se pueden dar tres casos:

2.1. Que los dos caracteres estén en la misma fila de la matriz de sustitución, en cuyo caso se sustituyen por los dos caracteres inmediatamente siguientes en la fila.

2.2. Si están en la misma columna, se sustituyen por los siguientes en la columna de la matriz.

2.3. Si no están en la misma fila ni en la misma columna, se sustituyen por los que forman un rectángulo con ellos en la matriz; es decir, los caracteres C_{ij} C_{kl} se sustituyen por los caracteres C_{il} C_{kj} de la matriz de sustitución. Por tanto, en nuestro caso en grupos de cinco caracteres, sería:

TEXTO CIFRADO: DTPED TZGLN DUBHR Y

que ha sido obtenido como sigue:

el primer grupo de letras ES que cumplen la condición 2.3 se sustituyen por DT:

DE
CF
KL
ST

el segundo grupo de letras, TO, también cumple la condición 2.3, por lo que se sustituyen por PE:

O R D E
A B C F
HIJ K L
P Q S T

vemos aquí que en este método la I y la J ocupan una misma posición.

El tercer grupo de letras, ES, es igual al primer grupo, por lo que se sustituye igualmente el cuarto grupo; UN cumple la condición 2.2, por lo que se sustituyen por los caracteres inmediatamente siguientes, ZG:

N
G
M
U
Z

el quinto grupo, ME, cumple la condición 2.3, por lo que se sustituyen por LN:

EN
FG
LM

el sexto grupo, NS, cumple la condición 2.3, por lo que son sustituidas por DU:

DEN
CFG
KLM
STU

el séptimo grupo, AJ, cumple la condición 2.3, por lo que se sustituyen por BH:

A B
HIJ

y por último, el grupo octavo, EW, cumple la condición 2.3, se sustituye por RY:

R D E
B C F
IJ K L
Q S T
W X Y



SISTEMAS HIBRIDOS

Los dos métodos básicos de sustitución y transposición pueden combinarse, dando lugar a unos sistemas más complejos. Así, se obtiene, por ejemplo, el sistema fraccionante, que consta de los siguientes pasos:

1. Sustitución bilateral. Transformando cada carácter en dos caracteres según una matriz, que contiene una palabra clave; supongamos que es CELDAS:

	1	2	3	4	5	6
1	C	E	L	D	A	S
2	B	F	G	H	I	J
3	K	M	N	O	P	Q
4	R	T	U	V	W	X
5	Y	Z	1	2	3	4
6	5	6	7	8	9	0

Así, el mensaje EL DIA 12 A LAS 14, ordenándolo en dos filas, en la primera el identificativo de la fila y en la segunda el identificativo de la columna, quedaría:

1 1 1 2 1 5 5 1 1 1 1 5 5

2 3 4 5 5 3 4 5 3 5 6 3 6

2. Transposición. Estas dos filas se transforman en una, concatenándolas de izquierda a derecha y de arriba abajo, quedando:

1 1 1 2 1 5 5 1 1 1 1 5 5 2 3 4 5 5 3 4 5 3 5 6 3 6

3. Sustitución. Según la matriz original, y tomando los números de dos en dos, quedaría:

TEXTO CIFRADO: CEAYCAZO3O44Q

Un programa que cifra-descifra según este método es el siguiente:

```
10 REM METODO HIBRIDO DE CIFRADO Y DESCIFRADO
15 LET A=64
20 DIM P(25):DIM L(25)
30 CLS:REM <-- EN EL COMMODORE CAMBIARLO POR: PRIN
T "[SHIFT-HOME]"
40 LOCATE 1,1:PRINT "METODO HIBRIDO DE CIFRADO Y D
ESCIFRADO"
50 LOCATE 20,1:PRINT"PARA FINALIZAR PULSE F"
60 LOCATE 3,1:INPUT "MODO (C/D : ",N$
70 IF N$="F" THEN GOTO 900
80 IF N$<"A" OR N$>"Z" THEN GOTO 60
90 LOCATE 5,1:INPUT "CLAVE : ";C$
100 LET LC=LEN(C$)
110 LET O$="N":LET I=1
118 IF MID$(C$,I,1)<"A" OR MID$(C$,I,1)>"Z" THEN L
ET O$="S"
120 LET I=I+1
122 IF I<=LC AND O$<>"S" THEN GOTO 118
125 IF O$="S" THEN GOTO 90
130 LET P=0
140 FOR I=1 TO LC
150 LET C=ASC(MID$(C$,I,1))-A:REM <-- EN EL SPECTR
UM CAMBIARLO POR: LET C=CODE(C$(I))-A
160 IF L(C)<=0 THEN LET P=P+1:LET P(P)=C:LET L(C)=
P
200 NEXT I
210 FOR I=1 TO 25
220 IF L(I)<=0 THEN LET P=P+1:LET P(P)=I:LET L(I)=
P
260 NEXT I
270 LOCATE 3,1:PRINT "        MODO = ";N$
280 LOCATE 5,1:PRINT "       CLAVE = ";C$
290 LOCATE 7,1:PRINT "ALFABETO ALTERADO = ";
300 FOR I=0 TO 4
310 FOR J=1 TO 5
320 PRINT CHR$(P(I*5+J)+A);
330 NEXT J
340 NEXT I
345 PRINT
350 INPUT "LONGITUD DEL BLOQUE : ";LB
```

```
370 IF LB<1 OR LB<>INT(LB) THEN GOTO 350
390 DIM B(2*LB)
400 LOCATE 11,1:INPUT "MENSAJE : ";M$
420 REM LIMPIA ESPACIOS DEL MENSAJE
430 GOSUB 5000
530 LOCATE 11,1: PRINT "    MENSAJE = ";M$
540 IF N$="C" THEN GOSUB 2000:GOTO 860
550 GOSUB 3000
860 LOCATE 20,1:PRINT "DESEA CONTINUAR (S/N)       :
"
870 LET R$=INKEY$:IF R$="" THEN GOTO 870:REM <-- E
N EL COMMODORE CAMBIARLO POR: GET R$:IF R$="" THEN
 GOTO 870
880 IF R$<>"N" AND R$<>"S" THEN GOTO 870
890 IF R$="S" THEN RUN
900 REM FIN
910 END:REM <-- EN EL SPECTRUM CAMBIARLO POR: STOP
2000 REM CIFRADO
2010 PRINT "MENSAJE CIFRADO = ";
2020 FOR I=1 TO LM STEP LB
2030 LET M=LB-1
2040 IF I+M>LM THEN LET M=LM-I
2050 FOR J=0 TO M
2060 LET L=ASC(MID$(M$,I+J,1))-A:REM <-- EN EL SPE
CTRUM CAMBIARLO POR: LET L=CODE(M$(I+J))-A
2070 LET P=L(L)
2080 LET B(J+1)=INT((P-1)/5)+1
2090 LET B(J+M+2)=P-5*(B(J+1)-1)
2100 NEXT J
2110 FOR J=0 TO 2*M STEP 2
2120 LET C=(B(J+1)-1)*5+B(J+2)
2130 PRINT CHR$(P(C)+A);
2140 NEXT J
2150 NEXT I
2160 RETURN
3000 REM DESCIFRADO
3010 PRINT "MENSAJE DESCIFRADO = ";
3020 FOR I=1 TO LM STEP LB
3030 LET M=LB-1
3040 IF I+M>LM THEN LET M=LM-I
3050 FOR J=0 TO M
3060 LET C=ASC(MID$(I+J),1))-A:REM <-- EN EL SPECT
RUM CAMBIARLO POR: LET C=CODE(M$(I+J))-A
3070 LET P=L(C)
3080 LET B(J*2+1)=INT((P-1)65)+1
3090 LET B(J*2+2)=P-5*(B(J*2+1)-1)
3100 NEXT J
3110 FOR J=0 TO M
```

```
3120 LET L=(B(J+1)-1)*5+B(J+M+2)
3130 PRINT CHR$(P(L)+A);
3140 NEXT J
3150 NEXT I
3160 RETURN
5000 REM LIMPIA TODOS LOS CARACTERES QUE NO SEAN L
ETRAS DEL MENSAJE
5010 LET I=2
5030 IF MID$(M$,I,1)<"A" OR MID$(M$,I,1)>"Z" THEN
LET M$=LEFT$(M$,I-1)+RIGHT$(M$,LEN(M$)-I):LET I=I-
1
5040 LET I=I+1
5050 IF I<LEN(M$) THEN GOTO 5030
5060 LET LM=LEN(M$)
5070 RETURN
```

COMENTARIOS DEL PROGRAMA HIBRIDO

Inicialización (10-20). Se definen las constantes y las tablas que empleará el programa.

Cabecera y mensaje de finalización (40-50). Imprime el nombre del método de cifrado y descifrado y a continuación imprime una línea indicando la forma de finalizar el programa (pulsando la tecla «F»).

Lectura del modo (60-80). Pide el modo de funcionamiento del programa; éste puede ser «C» para cifrado y «D» para descifrado. Si se pulsa la tecla «F», el programa finaliza. Si la tecla que se pulsa no es ninguna de las anteriores, el programa vuelve a pedir el modo.

Lectura de la clave (90-125). Pide la clave del método. Esta clave debe ser una cadena de caracteres. Si la clave introducida tiene caracteres no válidos, que no pertenecen al alfabeto (A..Z), el programa vuelve a pedir la clave.

Preparación de la clave (130-260). Obtiene la transformación del alfabeto a partir de la clave.

Imprime modo y clave (270-280). Imprime el modo y la clave para visualizar la elección anterior.

Imprime alfabeto modificado (290-340). Imprime el alfabeto transformado por la clave.

Lectura del número de caracteres (350-390). Pide el número de caracteres que se han de transponer cada vez, comprobando que sea un número entero positivo.

Lectura del mensaje (400). Pide el mensaje para descifrar o cifrar.

Limpia mensaje (5000-5070). Esta rutina elimina todos los caracteres que no pertenecen al alfabeto, y devuelve la longitud del mensaje.

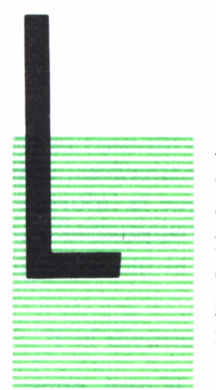
Selección en función del modo (540). Si el modo es «D», realiza el descifrado del mensaje. Si el modo es «C», realiza el cifrado.

Cifrado (2000-2160). Para cada uno de los caracteres del mensaje se le aplica la transformación, sustituyendo cada letra por las dos coordenadas que le direccionan en el cuadrado, se lee los dígitos por filas y a continuación se imprimen por columnas.

Descifrado (3000-3150). Para cada uno de los caracteres del mensaje se le aplica la transformación inversa, sustituyendo cada letra por las coordenadas que le direccionan en el cuadrado, leyendo los dígitos por columnas y a continuación se imprimen por filas.

Finalización o iteración (860-910). Pregunta al usuario si desea continuar; si la respuesta es «S», el programa vuelve al comienzo; si la respuesta es «N», finaliza. Si la respuesta no es ni «S» ni «N», repite la pregunta.

ESQUEMAS BASADOS EN EL COMPUTADOR 3



AS técnicas antes citadas pueden ser implementadas, como hemos visto, en un computador; sin embargo, en computadores estándar, los códigos máquina usados son fijos, y los datos deben estar en forma utilizable por la máquina, por lo que los símbolos cifrados deberán ser escogidos entre los que la máquina es capaz de utilizar, y así, no puede ser introducido cualquier símbolo cifrado.

Los lenguajes disponibles en el sistema también imponen una restricción, ya que los programas para la utilización de la criptografía deben estar codificados en algún lenguaje estándar, pues, en caso contrario, el coste de programación sería prohibitivo.

Los programas criptográficos deberán tener en cuenta las siguientes consideraciones:

a) La cantidad de confidencialidad decide el tiempo de computación y la labor de programación.

b) Las llaves usadas deben ser simples de construcción, fáciles de implementar y modificar en la máquina, y ocupar un espacio de memoria mínimo.

c) Los programas de cifrado y descifrado con llave conocida deben ser tan simples como sea posible y con un tiempo de computación pequeño.

d) Las llaves deben destruir los parámetros estadísticos y/o la estructura natural del lenguaje dado.

e) Los errores en el criptograma no deben causar ambigüedad o distorsiones en los datos originales, de forma que los hagan inservibles.

f) La capacidad de almacenamiento del criptograma no debe incrementar excesivamente la memoria.

g) El análisis del texto cifrado sin la llave deberá ser un problema de tal calibre que suponga un coste prohibitivo.

La aparición de los computadores dio origen a nuevos esquemas criptográficos. Entre los más importantes están los siguientes:



ESQUEMAS ARITMETICOS

Se basan en el hecho de que las operaciones aritméticas tienen la ventaja de ser fáciles de implementar. Entre éstos cabe destacar, respectivamente, los dos siguientes:



Suma y resta

Puesto que la adición y sustracción tienen operaciones inversas, que son, respectivamente, la sustracción y la adición, y puesto que la información en la memoria está representada de forma numérica, es posible utilizar estas operaciones para codificar datos, de forma que el mensaje cifrado C sea:

$$C = M \pm K$$

Siendo **M** el mensaje y **K** la llave, el mensaje descifrado M será, pues:

$$M = C \pm K$$

Veamos un ejemplo donde asignamos los siguientes valores a las letras del alfabeto:

Esp.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Ahora ciframos con el nuevo código la frase **ALTO SECRETO**, que nos da:

01122015001905031805201 5

a esto le añadimos la clave acordada con nuestro receptor, supongamos que es CONFIDENCIAL, que produce:

031514060904051403090112

sumando ahora las dos cifras obtenemos el siguiente valor:

```
  011220150019050318052015
+ 031514060904051403090112
  ------------------------
  042734210923101721142127
```

que convertido al alfabeto original nos produce el mensaje a enviar; previamente hemos hallado el mód. 26 de aquellas cifras que excedían de 26, así, enviaremos:

D GUIWJQUNU .

Hemos añadido un punto al final del mensaje para interpretar la última posición del mensaje, que es un espacio.



Multiplicación y división

Estas operaciones pueden ser utilizadas para transformar la información. Sin embargo, la multiplicación incrementa el tamaño del mensaje. En el caso de la división, además del divisor entero habrá que transmitir la parte fraccionaria de la división, o bien el resto.

El mensaje cifrado C será:

$$C = M \ {*}{\div} \ K$$

Y el mensaje descifrado M será:

$$M = C \ {\div}{*} \ K$$

La clave K, también puede tener la forma p ÷ q, donde p y q son enteros bien definidos.



Esquemas lógicos

Se basan en la propiedad que presentan las operaciones lógicas, o exclusivo, equivalencia y negación, de poseer operación inversa, y a que los computadores operan en código binario.

Suponiendo:

M = mensaje claro, K = clave y C = mensaje cifrado.

Así pues, en el caso del «0» exclusivo se tiene el cuadro siguiente:

⊕	0	1
0	0	1
1	1	0

que nos muestra cómo $M = K \oplus C$ y $C = M \oplus K$

Veamos un ejemplo:

MENSAJE CLARO:	01100011010111
CLAVE:	10010011100101
MENSAJE CIFRADO:	11110000110010

Para descifrarlo, basta con sumar al texto cifrado la clave

MENSAJE CIFRADO:	11110000110010
CLAVE:	10010011100101
MENSAJE CLARO:	01100011010111

En el caso de la equivalencia, se tiene el cuadro siguiente:

≡	0	1
0	1	0
1	0	1

que nos muestra cómo $M = C \equiv K$ y $C = M \equiv K$

Veamos un ejemplo:

MENSAJE CLARO:	100111000101
CLAVE:	011100000011
MENSAJE CIFRADO:	000100111001

para descifrarlo, basta con sumar al texto cifrado la clave:

MENSAJE CIFRADO:	000100111001
CLAVE:	011100000011
MENSAJE CLARO:	100111000101

En el caso de la negación, se tiene el cuadro siguiente:

–	
0	1
1	0

en esta operación no se necesitan dos operandos para efectuarla, solamente se cambian los 0 por 1 y los 1 por 0.



Esquemas matriciales

En este método, el mensaje M se descompone en elementos de una matriz rectangular de f filas y c columnas, de forma que si e son los elementos de M, entonces el número de elementos que tenga la matriz tendrá que ser igual o superior a e; en caso de que el número de elementos de la matriz sea superior, se rellenan los sobrantes con un elemento redundante.

A la matriz M se le puede sumar o multiplicar una matriz clave K. En el caso de la adición, el tamaño de K debe ser igual al de M, y se realizan menos operaciones que en el producto.

Si se utiliza la multiplicación, K debe tener una única inversa, por lo que la matriz K debe ser cuadrada no singular, de c*c elementos. En este caso, es preferible elegir un gran número de filas f para M y un número pequeño de columnas c. En el caso más simple, K puede ser escogida como una matriz ortogonal para que su inversa sea su transpuesta.

Supongamos que la asignación numérica al alfabeto es igual a la que se hizo para el ejemplo de la suma y resta, A = 01, B = 02, etc.

Si el mensaje que queremos enviar es: LLEGARE EL LUNES, lo que nos produce en un primer paso:

12120507011805000512001221140519

Lo ordenamos en forma de matriz de 4 por 4:

$$\begin{bmatrix} 12 & 12 & 05 & 07 \\ 01 & 18 & 05 & 00 \\ 05 & 12 & 00 & 12 \\ 21 & 14 & 05 & 19 \end{bmatrix}$$

Si lo multiplicamos por una matriz preestablecida:

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

nos produce (pasando a mód. 26):

$$\begin{bmatrix} 17 & 17 & 19 & 5 \\ 6 & 25 & 18 & 19 \\ 5 & 12 & 24 & 3 \\ 2 & 19 & 7 & 2 \end{bmatrix}$$

que en el alfabeto original es QQSEFYRSELXCBSGB.

Para descifrar el mensaje es necesario multiplicar por la inversa de la matriz preestablecida, y así obtendremos el mensaje claro.

TECNICAS AVANZADAS 4



A dijimos que existen dos tipos de algoritmos criptográficos: convencionales o de llave secreta y públicos. Con un algoritmo criptográfico convencional las llaves de cifrado y descifrado, o bien son idénticas, o si son diferentes, son de tal forma que una de ellas puede ser obtenida fácilmente a partir de la otra. En cambio, en un algoritmo de llave pública, muchos usuarios pueden codificar un texto según una llave pública de cifrado, pero sólo el usuario o destinatario específico del texto, o bien aquel que conoce la llave de descifrado puede decodificar dicho texto, ya que la llave de descifrado no se puede obtener a partir de la llave de cifrado.

A continuación se van a presentar tres ejemplos de algoritmos convencionales: el de llave en memoria, el de llave infinita y el D.E.S.; y dos algoritmos de llave pública, el R.S.A. y el método de Merkle y Hellman.



METODO DE LA LLAVE EN MEMORIA

Basado en la técnica desarrollada por Vernam, utiliza el «0» exclusivo. Skatrud, por su parte, implementó un sistema que utiliza dos llaves de memoria y una dirección de memoria. La sincronización se alcanza por medio de esta dirección, que es la primera información del texto. A partir de esta dirección, se localizan las posiciones de memoria que se utilizarán para la codificación. El proceso de modificación consta de dos partes:

a) $X = D \oplus k1$

b) $C = X \oplus k2$

Donde D representa el dato o carácter a codificar. K1 es la primera llave de memoria y K2 la segunda. A continuación se modifica el carácter siguiente utilizando otras llaves.

La seguridad del sistema depende de la cantidad de memoria usada, que, a su vez, depende de los mensajes.

Supóngase que se quieren transmitir mil mensajes de mil caracteres cada uno. Se necesitarán mil posiciones de memoria con objeto de no repetir ningún par de direcciones de memoria en las operaciones de cifrado. Entonces, el mensaje número 1 se codificará así:

Carácter	**Dirección de memoria 1**	**Dirección de memoria 2**
1	1	1
2	2	2
.	.	.
.	.	.
1000	1000	1000

A continuación el mensaje 2 ya no usaría los mismos pares de direcciones del mensaje 1 y se codificaría así:

Carácter	**Dirección de memoria 1**	**Dirección de memoria 2**
1	1	2
2	2	3
.	.	.
.	.	.
1000	1000	1

Y así sucesivamente, hasta el mensaje 1.000, que se iría codificando de la forma:

Carácter	**Dirección de memoria 1**	**Dirección de memoria 2**
1	1	1000
2	2	1
.	.	.
.	.	.
1000	1000	999

Por tanto, no se han repetido los pares de direcciones de memoria en la codificación de los 1.000 mensajes.

Para poder realizar el descifrado, las posiciones de memoria utilizadas no deben haber sufrido ningún cambio después de la codificación.



METODO DE LLAVE INFINITA

Mediante esta técnica, debida a Carrol y McLellan, se reduce la necesidad del almacenamiento de las llaves de memoria que utiliza el método anterior. Está basada en la generación de números aleatorios por el método de las congruencias, que está incorporado en cualquier computador moderno. Se procede de acuerdo con los siguientes pasos:

1. Se divide el texto a codificar en bloques de n caracteres.

2. Se escoge un valor inicial o semilla x0, para generar una serie de n números aleatorios.

3. Se genera una serie de números aleatorios x1, x2, ..., xn a partir del x0 y se efectúa el «0» exclusivo entre los caracteres ci del texto y los números xi obtenidos para los n caracteres del bloque del texto.

4. Si no existen más bloques a transmitir se termina el proceso. En caso contrario, se sigue en el punto siguiente.

5. Se escoge como nueva semilla x0, el último número aleatorio generado xn; es decir, x0 = xn y se vuelve al paso tercero.

Para poder descifrar el texto se procede de manera análoga, bastando con conocer la semilla original x0 para poder reconstruir el mensaje.



EL D.E.S. (DATA SCRYPTION STANDAR)



Génesis

De 1968 a 1975 un grupo de investigadores de la casa IBM (International Bussines Machines) estudiaron técnicas de cifrado para proteger, económica y eficazmente, la información almacenada en ficheros o transmitida por canales de comunicaciones. De estas investigaciones nació el sistema Lucifer, que utiliza combinaciones de transformaciones elementales simples tales como transposiciones fijas y sustituciones controladas por llaves.

En las aplicaciones comerciales, las especificaciones del algoritmo son públicas, hasta incluso normalizadas, residiendo todo el secreto en la llave. Además, los detalles del algoritmo deben elegirse de manera que permitan su implantación sobre un único circuito de larga escala de integración por aquella época. Con lo que resulta ser un dispositivo «hardware».

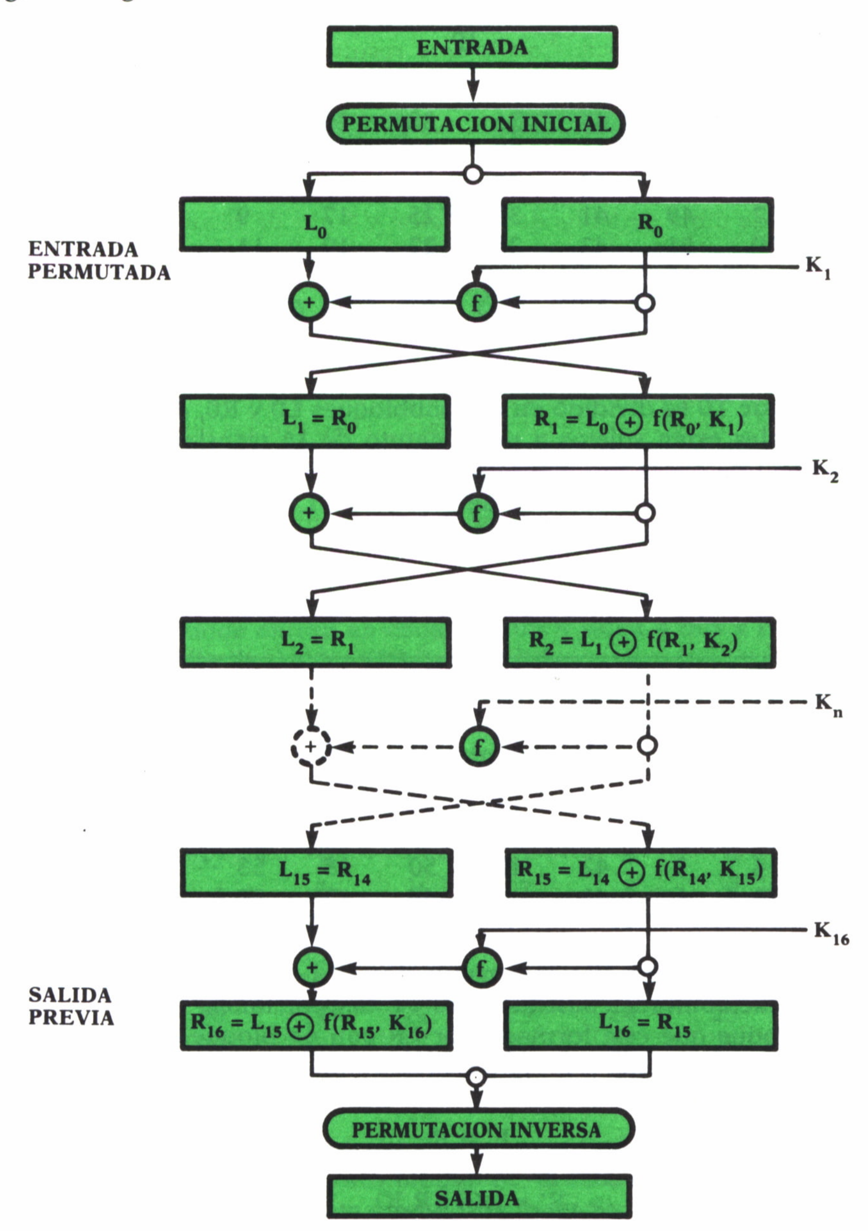
En el curso de estos estudios, la NSA (Agencia Nacional de Sefgural) de los EE.UU., se interesó en el desarrollo de esos trabajos. Las sustituciones son funciones no lineales realzadas con la ayuda de cajas S, que son una especie de tabla de 6 bits de dirección y 4 bits de resultado. En el bienio 1973-1974 la NSA «clasifica» los principios subyacentes en la elección de estas tablas, declarando que los investigadores habían redescubierto principios previamente clasificados.

En mayo de 1973 y en agosto de 1974 la NBS (Oficina Nacional de Estándares), sensibilizada por el contexto político, publica llamadas para la adquisición de algoritmos de cifrado para su uso por las distintas agencias federales. Entre los algoritmos remitidos, respondiendo a esta llamada, se encontraba el que envió IBM con el nombre de «Encrytion algorithm for computer data protection". Este algoritmo fue publicado en marzo de 1975 y en agosto del mismo año, con una petición de comentarios con el fin de establecer una norma de uso en las agencias federales. IBM había precisado en aquel momento que si el algoritmo propuesto era el seleccionado como una norma federal, renunciaría a las patentes implicadas en los límites territoriales de los EE.UU.

El 15 de julio de 1977 el algoritmo fue elegido y rápidamente bautizado como D.E.S., quedando sometido a control oficial, ejercido por la NSA, sobre la exportación de circuitos. Este control, impuesto en nombre de la seguridad, abocó a un proteccionismo exagerado de los productos y los servicios propuestos por las firmas norteamericanas. Este control «incordia» mucho, fuera de los EE.UU., el desarrollo de maquetas y experiencias dedicadas a precisar la puesta en marcha del cifrado en los protocolos de comunicaciones. Y, lo que es aún más grave, D.E.S., en principio elegido para su uso por las agencias federales, se ha convertido de hecho en una norma comercial. Así, en marzo de 1978 el American National Standards Institute (ANSI) eligió al D.E.S. como norma comercial de cifrado. Y el grupo de trabajo ANSIX3S3 estudia actualmente los problemas de introducción del cifrado en los protocolos de comunicación.

Aunque las investigaciones realizadas alrededor del D.E.S. y su existencia sean incluso obstáculos en el desarrollo y la puesta en marcha de un nuevo algoritmo concurrente, es absolutamente necesario el conocimiento técnico y la voluntad política e industrial para desarrollar en Europa un algoritmo de cifrado.

El D.E.S. consiste en un algoritmo de cifrado-descifrado de bloques de 64 bits, mediante una clave de 56 bits, cuyos pasos se detallan en la siguiente figura:



Los 64 bits de entrada forman un bloque T, que es transformado mediante la permutación IP, dando un bloque T0 = IP(T).

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Este bloque T0 es dividido en dos subbloques L0 y R0, de 32 bits cada uno, los cuales estan sujetos a un conjunto de 16 transformaciones, de acuerdo con una cierta función f y 16 subclaves Ki = (i = 1,...,16), al final de los cuales el bloque resultante de unir R16 a L16 es sometido a la permutación inversa de la inicial (IP-1):

IP-1

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	2
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Entre la permutación IP y la permutación IP-1, el algoritmo ejecuta 16 iteraciones que combinan sustituciones y transposiciones.

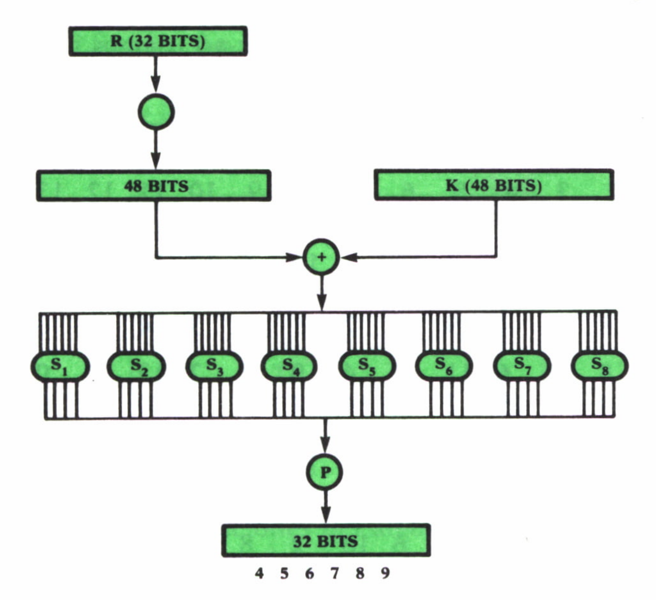
Así, el bloque que está formado por LR con un bloque de clave K de 48 bits producen la salida L'R', que está definido por:

$$L' = R$$

$$R' = L \oplus f(R,K)$$

siendo $\oplus$ la operación or-exclusivo.

La función f(R,K) se obtiene como muestra la siguiente figura:



Siendo E una función de expansión, la operación suma módulo 2, ocho funciones Si donde entran en cada S 6 bits y salen 4 bits y P una permutación final antes de dar los 32 bits de salida.

La función E con un bloque de 32 bits produce otro bloque de 48 bits; el proceso es que a partir de los 32 bits originales produce una permutación de ellos, produciendo el siguiente bloque:

E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Seguidamente a esta función E se le suma (mód. 2) K, que también posee 48 bits, obteniéndose otro bloque de 48 bits, que, dividido en grupos de 6 bits, hacen las entradas a las 8 funciones Si.

Estas funciones Si reciben un bloque de 6 bits (b1b2b3b4b5b6); se toman los bits b1 y b6 de tal forma que representan un número en base 2

(binario), cuyo rango está entre 0 y 3, este número lo denotamos por i. A continuación los bits b2b3b4b5 representan otro número en base 2 de rango 0 al 15, este número lo denotamos por j; así usamos i para las filas y j para las columnas de las 8 tablas siguientes:

	Columna																
Fila	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	4	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	5	4	3	
0	12	1	10	15	9	2	6	8	0	1 3	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	11	3	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	1 4	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	15	13	8	6	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Como vemos, el máximo valor que puede salir es 15, escrito en binario, 1111, con lo que nos valen 4 bits para su representación, y que son el tamaño que dan como salida las funciones Si.

Definamos ahora la permutación final Pi; ésta es una ordenación de los 32 bits que le llegan y lo hacen de esta forma:

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Nos falta por describir cómo opera la función de clave K; vamos a verlo.

Cada una de las 16 iteraciones del algoritmo D.E.S. utiliza una clave (Ki) diferente de 48 bits, la cual es hallada a partir de la clave original de 64 bits mediante las dos tablas siguientes:

PC-1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
61	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

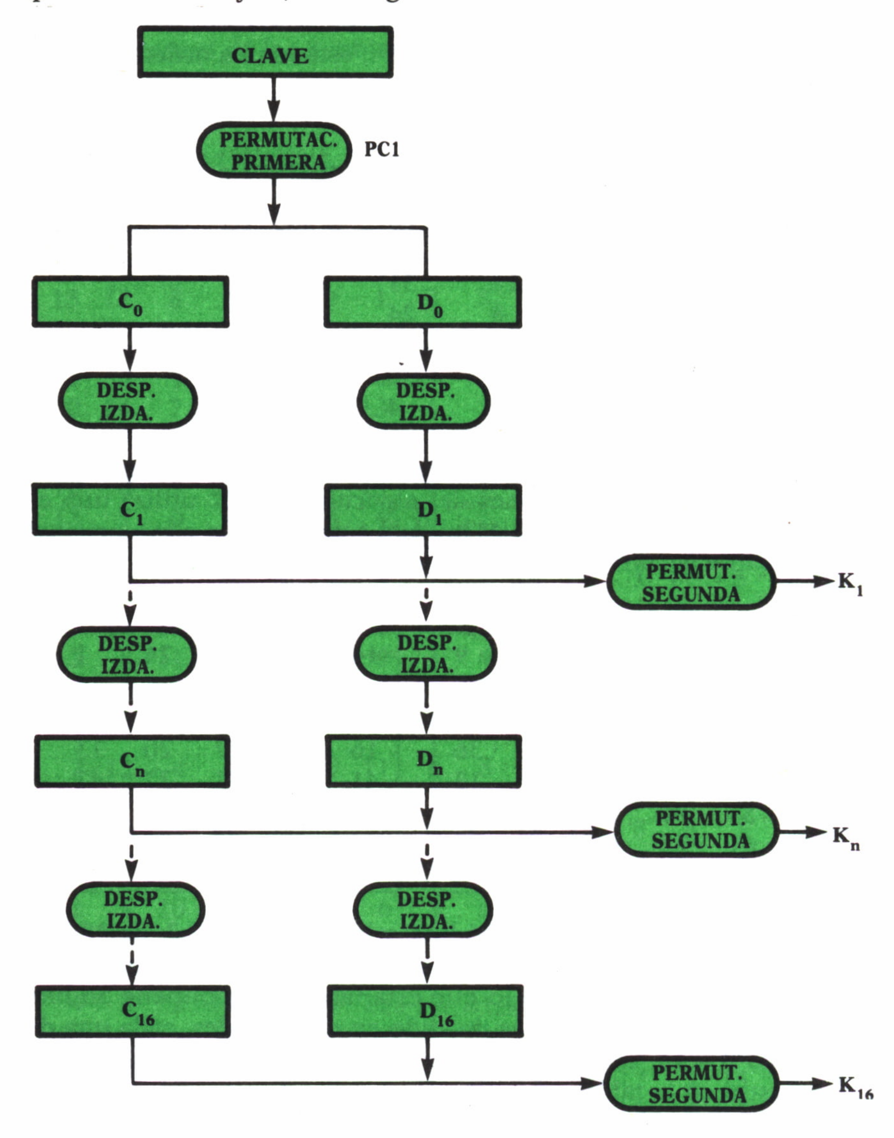
la primera tabla, PC-1, consta de 56 bits, y la segunda, PC-2, de 48 bits.

A la entrada de PC-1 de los 64 bits que llegan se desestiman los bits 8, 16, 24, 32, 40, 48, 56 y 64, con lo que esta tabla sólo contiene 56 bits; el resultado de esta permutación es dividido en dos mitades, cada una de 28 bits de rango 0 a 16, cada mitad sufre un desplazamiento a la izquierda, que se muestra en la tabla siguiente:

Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit a desplazar a la izquierda	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Después de este desplazamiento, concatenamos los bloques de las dos mitades y aplicamos la permutación PC-2, eliminando los bits de las posiciones 9, 18, 22, 25, 35, 38, 43 y 54, obteniendo así la clave Ki de 48 bits.

Un gráfico que nos muestra la obtención de Ki, denominando a cada bloque de 28 bits L y M, es el siguiente:



Lo expuesto hasta el momento es para el cifrado, pero dado que las permutaciones inicial y final son una inversa de la otra y aplicando las subclaves en orden inverso a como fueron obtenidas, obtendríamos el descifrado.

Un programa realizado en Turbo Pascal para MS-DOS y que realiza el cifrado y descifrado según el algoritmo D.E.S., es el siguiente:

```
program data_encryption_standard;
{$R+}
{-----------------------------------------------------------------------------

    Con objeto de ser utilizado para estudiar el algoritmo, al ejecutarse
  el programa se va ofreciendo una traza de los distintos pasos (rondas) por
  los que va evolucionando.

    El programa exige que la clave y el texto de entrada estén en dos ficheros,
  cuya identidad se pregunta al usuario. Estos ficheros contendrán la informa-
  ción en forma de '1' y '0'. Asimismo, se genera un fichero de salida con el
  texto cifrado/descifrado.


 -----------------------------------------------------------------------------}

type
    bit      = 0..1; { FALSE..TRUE }
    tira64   = array [1..64] of boolean;  { of bit }
    tira56   = array [1..56] of boolean;  { of bit }
    tira48   = array [1..48] of boolean;  { of bit }
    tira32   = array [1..32] of boolean;  { of bit }
    tira28   = array [1..28] of boolean;  { of bit }
    tira6    = array [1..6] of boolean;   { of bit }
    tira4    = array [1..4] of boolean;   { of bit }
    tira8x6 = array [1..8,1..6] of boolean; { of bit }
    orden64 = array [1..64] of integer;
    orden56 = array [1..56] of integer;
    orden48 = array [1..48] of integer;
    orden32 = array [1..32] of integer;
    matriz_4x16= array [0..3, 0..15] of integer;

var
   K: array [1..16] of tira48;
   L, R: array [0..16] of tira32;
   fclave, fentrada, fsalida: text;
   entrada, salida, clave: tira64;
   paso, i, j: integer;
   car: char;
   nombre_fichero: string [14];
{ ------------------ generación de claves ------------------------------ }
procedure generar_claves;  { genera el vector de claves K [1..16] }
var
    i: integer;
    j: integer;
    c, d: array [0..16] of tira28;

const ciclo: array [1..16] of integer = (1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1);

const PC_1: orden56 = (57,49,41,33,25,17, 9,
                        1,58,50,42,34,26,18,
                       10, 2,59,51,43,35,27,
                       19,11, 3,60,52,44,36,
                       63,55,47,39,31,23,15,
                        7,62,54,46,38,30,22,
                       14, 6,61,53,45,37,29,
                       21,13, 5,28,20,12, 4 );

const PC_2: orden48 = (14,17,11,24, 1, 5,
                        3,28,15, 6,21,10,
                       23,19,12, 4,26, 8,
                       16, 7,27,20,13, 2,
                       41,52,31,37,47,55,
                       30,40,51,45,33,48,
```

```
                      44,49,39,56,34,53,
                      46,42,50,36,29,32 );

procedure PC1; { devuelve c [0] y d [0] }
  var i: integer;
  begin
       for i:= 1 to 28 do
          begin
               c [0][i] :=  clave [ PC_1 [ i ] ];
               d [0][i] :=  clave [ PC_1 [ i + 28 ] ];
           end;
  end;

procedure PC2 (indice: integer);  { devuelve K [indice] }
  var i: integer;
      cd: tira56;
  begin
       for i:= 1 to 28 do
          begin
               cd [ i ]:= c [ indice ] [i];
               cd [ i+28 ]:= d [ indice ] [i];
          end;
       for i:= 1 to 48 do K [indice][i]:= cd [ PC_2 [i] ];
  end;
 procedure desplazar_c (nro, indice: integer);
   var i: integer;
   begin
     for i:= 1 to 28 do c [indice][i]:= c [indice-1] [(i-1 + nro) mod 28 + 1]
   end;

 procedure desplazar_d (nro, indice: integer);
   var i: integer;
   begin
     for i:= 1 to 28 do d [indice][i]:= d [indice-1] [(i-1+ nro) mod 28 + 1]
   end;


  begin  { generar_claves }
        PC1;
        for i:= 1 to 16 do
          begin
               desplazar_c (ciclo [i], i);
               desplazar_d (ciclo [i], i);
               PC2 (i);
               write ('K [',i:2,']: ');
               for j:= 1 to 48 do if K [i][j] then write ('1') else write ('0');
               writeln
          end
 end;  { ------------------ generar_claves -------------------------------- }
 procedure permutacion_inicial (indice: integer);
  { -------- obtiene los valores iniciales de L [] y R [] -------- }
   const pi: orden64 = (58,50,42,34,26,18,10, 2,
                        60,52,44,36,28,20,12, 4,
                        62,54,46,38,30,22,14, 6,
                        64,56,48,40,32,24,16, 8,
                        57,49,41,33,25,17, 9, 1,
                        59,51,43,35,27,19,11, 3,
                        61,53,45,37,29,21,13, 5,
                        63,55,47,39,31,23,15, 7 );
   var i: integer;
   begin
      for i:= 1 to 32 do if indice = 0 then begin
                                              L [0][i]:= entrada [ pi [i] ];
                                              R [0][i]:= entrada [ pi [i+32] ]
                                            end
                                       else begin
                                              R [16][i]:= entrada [ pi [i] ];
                                              L [16][i]:= entrada [ pi [i+32] ]
                                            end
  end;  { ----------------- permutación_inicial --------------------------- }
```

```
procedure permutacion_final (indice: integer);
 { -------- obtiene la salida a partir de L [] y R [] -------- }
  const pf: orden64 = (40, 8,48,16,56,24,64,32,
                       39, 7,47,15,55,23,63,31,
                       38, 6,46,14,54,22,62,30,
                       37, 5,45,13,53,21,61,29,
                       36, 4,44,12,52,20,60,28,
                       35, 3,43,11,51,19,59,27,
                       34, 2,42,10,50,18,58,26,
                       33, 1,41, 9,49,17,57,25 );
  var i: integer;
      rl: tira64;
  begin
     for i:= 1 to 32 do if indice = 0 then begin
                                          rl [ i ] :=  L [0][i];
                                          rl [ i+32 ] :=  R [0][i]
                                        end
                                  else begin
                                          rl [ i ] :=  R [16][i];
                                          rl [ i+32 ] :=  L [16][i]
                                        end;
     for i:= 1 to 64 do salida [i]:= rl [ pf [i] ];

 end;  { ------------------- permutación_final --------------------------- }

{**** La relación entre pf y pi es la siguiente:

                     for i:= 1 to 64 do  pf [ pi [i] ] := i

 ******************************************************************* }
const
    s: array [1..8] of matriz_4x16 =
          { s1 }     (((14, 4,13, 1, 2,15,11, 8, 3,10, 6,12, 5, 9, 0, 7),
                       ( 0,15, 7, 4,14, 2,13, 1,10, 6,12,11, 9, 5, 3, 8),
                       ( 4, 1,14, 8,13, 6, 2,11,15,12, 9, 7, 3,10, 5, 0),
                       (15,12, 8, 2, 4, 9, 1, 7, 5,11, 3,14,10, 0, 6,13) ),

          { s2 }      ((15, 1, 8,14, 6,11, 3, 4, 9, 7, 2,13,12, 0, 5,10),
                       ( 3,13, 4, 7,15, 2, 8,14,12, 0, 1,10, 6, 9,11, 5),
                       ( 0,14, 7,11,10, 4,13, 1, 5, 8,12, 6, 9, 3, 2,15),
                       (13, 8,10, 1, 3,15, 4, 2,11, 6, 7,12, 0, 5,14, 9) ),

          { s3 }      ((10, 0, 9,14, 6, 3,15, 5, 1,13,12, 7,11, 4, 2, 8),
                       (13, 7, 0, 9, 3, 4, 6,10, 2, 8, 5,14,12,11,15, 1),
                       (13, 6, 4, 9, 8,15, 3, 0,11, 1, 2,12, 5,10,14, 7),
                       ( 1,10,13, 0, 6, 9, 8, 7, 4,15,14, 3,11, 5, 2,12) ),

          { s4 }      (( 7,13,14, 3, 0, 6, 9,10, 1, 2, 8, 5,11,12, 4,15),
                       (13, 8,11, 5, 6,15, 0, 3, 4, 7, 2,12, 1,10,14, 9),
                       (10, 6, 9, 0,12,11, 7,13,15, 1, 3,14, 5, 2, 8, 4),
                       ( 3,15, 0, 6,10, 1,13, 8, 9, 4, 5,11,12, 7, 2,14) ),

          { s5 }      (( 2,12, 4, 1, 7,10,11, 6, 8, 5, 3,15,13, 0,14, 9),
                       (14,11, 2,12, 4, 7,13, 1, 5, 0,15,10, 3, 9, 8, 6),
                       ( 4, 2, 1,11,10,13, 7, 8,15, 9,12, 5, 6, 3, 0,14),
                       (11, 8,12, 7, 1,14, 2,13, 6,15, 0, 9,10, 4, 5, 3) ),

          { s6 }      ((12, 1,10,15, 9, 2, 6, 8, 0,13, 3, 4,14, 7, 5,11),
                       (10,15, 4, 2, 7,12, 9, 5, 6, 1,13,14, 0,11, 3, 8),
                       ( 9,14,15, 5, 2, 8,12, 3, 7, 0, 4,10, 1,13,11, 6),
                       ( 4, 3, 2,12, 9, 5,15,10,11,14, 1, 7, 6, 0, 8,13) ),

          { s7 }      (( 4,11, 2,14,15, 0, 8,13, 3,12, 9, 7, 5,10, 6, 1),
                       (13, 0,11, 7, 4, 9, 1,10,14, 3, 5,12, 2,15, 8, 6),
                       ( 1, 4,11,13,12, 3, 7,14,10,15, 6, 8, 0, 5, 9, 2),
                       ( 6,11,13, 8, 1, 4,10, 7, 9, 5, 0,15,14, 2, 3,12) ),

          { s8 }      ((13, 2, 8, 4, 6,15,11, 1,10, 9, 3,14, 5, 0,12, 7),
                       ( 1,15,13, 8,10, 3, 7, 4,12, 5, 6,11, 0,14, 9, 2),
```

```
                    ( 7,11, 4, 1, 9,12,14, 2, 0, 6,10,13,15, 3, 5, 8),
                    ( 2, 1,14, 7, 4,10, 8,13,15,12, 9, 0, 3, 5, 6,11) ));

expansion: orden48 = (32, 1, 2, 3, 4, 5,
                       4, 5, 6, 7, 8, 9,
                       8, 9,10,11,12,13,
                      12,13,14,15,16,17,
                      16,17,18,19,20,21,
                      20,21,22,23,24,25,
                      24,25,26,27,28,29,
                      28,29,30,31,32, 1 );

P : orden32 = (16, 7,20,21,
               29,12,28,17,
                1,15,23,26,
                5,18,31,10,
                2, 8,24,14,
               32,27, 3, 9,
               19,13,30, 6,
               22,11, 4,25 );

 procedure cifrado; { ------------- procedimiento de cifrado -------------- }

 procedure funcion_cifrado (indice: integer);
 {  obtiene R [indice] := L [indice - 1] xor F (R [indice - 1], K [indice])  }
  var i: integer;
      F: tira32;
  procedure calcular_F;
   var a, h, i, j: integer;
       suma      : tira8x6;
       exp_R     : tira48;
       comp32    : tira32;
   begin
        for i:= 1 to 48 do exp_R [ i ] := R [indice-1] [expansion [i]];
        for i:= 0 to 7 do
           for j:= 1 to 6 do suma [i+1,j]:=  exp_R [i*6+j]
                                              xor K [indice][i*6+j];
        for h:= 1 to 8 do
           begin
             i:= ord (suma [h,6]) + 2 * ord (suma [h,1]);
             j:= ord (suma [h,5]) + 2 * ord (suma [h,4])
                 + 4 * ord (suma [h,3]) + 8 * ord (suma [h,2]);
             a:= s [h] [i,j];
             comp32 [(h-1)*4 + 1]:= boolean ( a div 8 );  a:= a mod 8;
             comp32 [(h-1)*4 + 2]:= boolean ( a div 4 );  a:= a mod 4;
             comp32 [(h-1)*4 + 3]:= boolean ( a div 2 );  a:= a mod 2;
             comp32 [(h-1)*4 + 4]:= boolean ( a )
           end;
        for i:=1 to 32 do F [i]:= comp32 [P [i]];
   end; { calcular_F }
 begin
      calcular_F;
      for i:=1 to 32 do R [indice][i]:=  L [indice-1][i]  xor  F [i];

 end; {  funcion_cifrado ( indice )  }

 begin  { cifrado }
       write ('> ENTRADA: ');
       for i:= 1 to 32 do if entrada [i] then write ('1') else write ('0');
       write (' ');
       for i:= 33 to 64 do if entrada [i] then write ('1') else write ('0');
       writeln;

       permutacion_inicial (0);
       write ('L[ 0]R[ 0]:');
       for i:= 1 to 32 do if L[0][i] then write ('1') else write ('0');
       write (' ');
       for i:= 1 to 32 do if R[0][i] then write ('1') else write ('0');
       writeln;
```

```
      for paso:= 1 to 16 do
         begin
          L [paso] := R [paso - 1];
          funcion_cifrado (paso);
         { R [paso] := L [paso - 1] xor F (R [paso - 1], K [paso])); }
          write ('L[',paso:2,']R[', paso:2,']:');
          for i:=1 to 32 do if L [paso] [i] then write ('1') else write ('0');
          write (' ');
          for i:=1 to 32 do if R [paso] [i] then write ('1') else write ('0');
          writeln
         end;
       permutacion_final (16);

end; { ------------------------- cifrado ---------------------------------- }

procedure descifrado; { ----------- procedimiento de descifrado ----------- }

procedure funcion_descifrado (indice: integer);
{  obtiene L [indice-1] := R [indice] xor F (L [indice], K [indice])  }
var i: integer;
    F: tira32;
procedure calcular_F;
  var a, i, j, h: integer;
      suma: tira8x6;
      exp_L: tira48;
      comp32: tira32;
  begin
       for i:= 1 to 48 do exp_L [ i ] := L [indice] [expansion [i]];
       for i:= 0 to 7 do
          for j:= 1 to 6 do suma [i+1,j]:=  exp_L [i*6+j]
                                            xor K [indice][i*6+j];
       for h:= 1 to 8 do
         begin
           i:= ord (suma [h,6]) + 2 * ord (suma [h,1]);
           j:= ord (suma [h,5]) + 2 * ord (suma [h,4])
               + 4 * ord (suma [h,3]) + 8 * ord (suma [h,2]);
           a:= s [h] [i,j];
           comp32 [(h-1)*4 + 1]:= boolean ( a div 8 );  a:= a mod 8;
           comp32 [(h-1)*4 + 2]:= boolean ( a div 4 );  a:= a mod 4;
           comp32 [(h-1)*4 + 3]:= boolean ( a div 2 );  a:= a mod 2;
           comp32 [(h-1)*4 + 4]:= boolean ( a )
         end;
       for i:=1 to 32 do F [i]:= comp32 [P [i]];
  end; { calcular_F }
begin  {  funcion_descifrado  }
   calcular_F;
   for i:=1 to 32 do L [indice-1][i]:=  R [indice][i]  xor  F [i];

end; {  funcion_descifrado ( indice )  }

begin { descifrado }
      write ('> ENTRADA: ');
      for i:= 1 to 32 do if entrada [i] then write ('1') else write ('0');
      write (' ');
      for i:= 33 to 64 do if entrada [i] then write ('1') else write ('0');
      writeln;

      permutacion_inicial (16);
      write ('R[16]L[16]:');
      for i:= 1 to 32 do if R[16][i] then write ('1') else write ('0');
      write (' ');
      for i:= 1 to 32 do if L[16][i] then write ('1') else write ('0');
      writeln;

      for paso:= 15 downto 0 do
         begin
          R [paso] := L [paso+1];
          funcion_descifrado (paso+1);
         { L [paso] := R [paso + 1] xor F (L [paso + 1], K [paso+1])); }
          write ('R[',paso:2,']L[', paso:2,']:');
```

```
        for i:=1 to 32 do if R [paso] [i] then write ('1') else write ('0');
        write (' ');
        for i:=1 to 32 do if L [paso] [i] then write ('1') else write ('0');
        writeln
       end;
     permutacion_final (0);
end; { ---------------------- descifrado -------------------------------- }

begin { programa principal }
  clrscr;
  write ('Nombre del fichero clave: '); readln (nombre_fichero);
  assign (fclave, nombre_fichero);
  {$I-} reset (fclave) {$I+};
  if IOresult <> 0 then writeln ('>>> ERROR: no se puede abrir el fichero ',
                              nombre_fichero)
  else
    begin { continua el programa principal }
      for i:= 1 to 64 do begin read (fclave, car);
                             if car='1' then clave [i]:= TRUE
                                        else clave [i]:= FALSE;
                       end;

      write ('CLAVE: ');
      for i:= 1 to 64 do if clave [i] then write ('1') else write ('0');
      writeln;
      close (fclave);
      generar_claves;

      writeln;
      write ('Nombre del fichero de entrada: '); readln (nombre_fichero);
      assign (fentrada, nombre_fichero);
      {$I-} reset (fentrada) {$I+};
      if IOresult <> 0 then
        writeln ('>>> ERROR: no se puede abrir el fichero ',nombre_fichero)
      else
        begin { continua el programa principal }
          for i:= 1 to 64 do begin read (fentrada, car);
                                 if car='1' then entrada [i]:= TRUE
                                            else entrada [i]:= FALSE;
                           end;


          write ('Nombre del fichero de salida: '); readln (nombre_fichero);
          assign (fsalida, nombre_fichero);
          rewrite (fsalida);

          write ('Desea cifrar o descifrar ? (c/d): '); readln (car);
          if car in ['c','C','d','D'] then begin
          case car of
                 'c','C': cifrado;
                 'd','D': descifrado;
                 else writeln ('Si usted lo desea, no se realizará nada');
          end;

           for i:= 1 to 64 do if salida [i] then write (fsalida,'1')
                                            else write (fsalida,'0');
           write ('> SALIDA:  ');
           for i:= 1 to 32 do if salida [i] then write ('1') else write ('0');
           write (' ');
           for i:= 33 to 64 do if salida [i] then write ('1') else write ('0');
           writeln;
           end
           else
            writeln ('>> Opción incorrecta: no se realiza ninguna operación');
           close (fentrada);
           close (fsalida);
        end
    end
 end.
```



La problemática D.E.S.

Existen tres argumentos principales contra el uso del D.E.S. que aquí se van a describir someramente. El primero, de gran importancia, es que se guardan cosas en secreto. Por ejemplo, hay quien dice que, con la disculpa de verificar el algoritmo, la NSA lo cambió con el fin de reducirlo y crear una «trampa» de la que sólo la NSA tiene la llave. Otros afirman que la NSA forzó a IBM a mantener secreto el diseño de las cajas S, de modo que ésta puede ser la «trampa». Otros arguyen que fue la NSA la que obligó a IBM a reducir el tamaño de la llave de 128 a 56 bits, y justo ahí puede estar parte de la «trampa».

Ante estos alegatos, tanto IBM como la NSA se declaran inocentes de cualquier tipo de manipulaciones secretas; sin embargo, muchos suponen que, efectivamente, hay «gato encerrado», por lo que difícilmente puede considerarse a D.E.S. como de llave pública.

El segundo argumento contra D.E.S. afirma que la llave de 56 bits es demasiado corta. En efecto, Diffie y Hellman, investigadores de Stanford, señalaron la posibilidad de un ataque brutal contra D.E.S., gracias a una máquina especializada que tendría un millón de bits, cada uno de ellos diseñado para probar un millón de claves por segundo. Según estos investigadores, con esta máquina se podría descifrar un mensaje en clave en pocas horas y a un coste medio de solución de 5.000 dolares. Esta máquina, evaluada por ellos en 20.000.000 de dolares, es fácil de construir y su coste disminuirá de año en año. Estas ideas fueron consideradas como poco realistas en el momento de su publicación y no provocaron ninguna reacción oficial, tanto más cuanto que los oficiales de NBS intentaban desacreditar en privado a estos teóricos de la información y, por elevación, sus críticas al asunto D.E.S. Pero en 1977 Rivest, basándose en estos trabajos de Diffie y Hellman, publicó sendos artículos en las revistas *Science* y *Scientific American,* en donde explicaba detalladamente sus trabajos recogidos en la memoria técnica 82 de abril de 1977 de LCS (Laboratorio of Computer Science) del MIT. Con lo que las nuevas ideas sobre criptografía no estaban confinadas al «ghetto» de sólo los iniciados. A partir de ese momento, las ideas de Diffie y Hellman dejaron de ser irrealistas y desde entonces la NSA se interesa en el desarrollo de cifrados de llave pública.

El tercer argumento es que justamente por esas razones los sistemas de verdadera llave pública son mejores que D.E.S. Estos sistemas de cifrado son los que se van a considerar a continuación.



ALGORITMOS DE LLAVE PUBLICA



Bases teóricas

En el curso del Congreso sobre la Teoría de la Información, celebrado en junio de 1975 en Lenox, Massachusetts, Diffie y Helllman presentaron ideas originales sobre sistemas de cifrado. Allí pusieron de evidencia el problema fundamental de las redes públicas de transmisión de información; a saber, la distribución de llaves secretas y la identificación entre corresponsales, sugiriendo métodos nuevos para alcanzar estos resultados.

Sus reflexiones se apoyan en la teoría de la complejidad del cálculo. Para evaluar una función en un punto se puede bien calcular directamente el valor usando un algoritmo adecuado, o bien buscando en una tabla de valores precalculados. De este modo, la complejidad del cálculo puede expresarse como un compromiso entre el número de operaciones elementales a efectuar; es decir, el tiempo y la energía consumidos en el cálculo y el tamaño de la memoria necesario, o espacio de cálculo empleado. Para un problema dado, el mejor algoritmo es el que minimiza el producto de ambos factores.

La noción de función prácticamente no invertible se deduce de estas consideraciones y es relativa al estado actual de los conocimientos de los algoritmos, así como al estado actual de la tecnología que define la potencia de cálculo disponible. Para que sea utilizable, una función prácticamente no invertible debe ser fácil de calcular, mientras que el cálculo de su función inversa necesita un tiempo astronómico y una memoria galáctica.

Y, justamente, ciertos problemas matemáticos no pueden resolverse actualmente más que por métodos demasiado lentos, incluso por los computadores más potentes. Claro que el hecho de que no se conozcan métodos eficientes no quiere decir que no existan.

En noviembre de 1976 Diffie y Hellman fueron los primeros en imaginar funciones de cambio de llaves que permiten establecer una llave secreta entre dos corresponsales ligados por un canal de comunicación de escucha pasiva. Siendo asimismo los primeros en definir las propiedades teóricas de los sistemas de cifrado con llave pública que permiten el equivalente electrónico de las firmas, algo que no permite el D.E.S.

Estas funciones de cambio de llave se basan en ciertos resultados clásicos de la aritmética que se mostraron adecuados a las nuevas necesidades, como el siguiente: sea P un número primo, las leyes de la suma y de la multiplicación módulo 7 dotan al conjunto de los enteros de 0 a P-1 de una estructura de cuerpo. Es el cuerpo de Galois CG(P). Siendo dados x e

y dos elementos no nulos de CG(P), se puede definir x ** y módulo P como la potencia y-ésima de x módulo P. Si x es un elemento primitivo; es decir, generando por sus potencias sucesivas de todos los elementos no nulos del cuerpo, entonces la transformación de y en z, tal que z = x ** y módulo P se denomina «exponencial», mientras que la transformación inversa se denomina «logaritmo de base x sobre GC(P)».

Si P y (P-1)/2 son dos números primos, entonces la exponencial sobre GC(P) es una función prácticamente no invertible. Si el número P se escribe sobre n bits, el cálculo de la exponencial se hace con tres palabras de n bits en un tiempo proporcional a n**3, en tanto que el cálculo de un algoritmo por el mejor algoritmo conocido se hace con una memoria y un tiempo proporcional a 2**(n2). Números primos de esta forma existen y se localizan bastante fácilmente. Incluso se pueden elegir de manera que se puedan simplificar los cálculos en binario. Por ejemplo, 2210-65 y 2209-33 cumplen estos requisitos.

Una propiedad de base de las funciones exponenciales es que:

$$(a ** x) y = (a ** y) x = a ** xy$$

lo que permite hacer de la exponencial sobre CG(P) una función de cambio de llaves, y concebir a continuación un servicio público de identificación y discreción sobre una red pública. La forma de hacer esto es como sigue: la administración retiene un gran número primo P, por ejemplo, 2209-65, y genera periódicamente al azar un elemento primitivo de CG(P), por ejemplo «e». Luego indica estos valores P y e a cada usuario que posee una caja negra como interfaz con la red. De este modo, un usuario del servicio, tal que U1, genera al azar y secretamente un número impar «i» de 210 bits e introduce este valor en su caja negra como identificador secreto. Después calcula, también en secreto, la imagen pública de este identificador como sigue:

$$A = e ** i \bmod P$$

que constituye su identificador público, que transmite a sus principales «interlocutores» y a la administración para que lo inserte en el «armario» público. Este «armario», establecido y gestionado bajo la responsabilidad de la administración, indicará además del nombre de los usuarios y del número de acceso a la red, su identificador público. Eventualmente podrá ser consultado a través de la red por un acceso particular. Entonces, el acceso estará provisto de una caja negra parecida a la de los usuarios, de modo que identifique el «armario» en el momento de las consultas, pues es esencial que dicho «armario» no pueda ser ni simulado ni falsificado.

Cuando dos usuarios, U1 y U2, desean establecer una comunicación secreta, combinan el identificador público de su interlocutor con su identificador secreto, con el fin de obtener la llave de conexión característica

de su comunicación durante el período de validez del elemento primitivo «e». Para U1 la llave es:

$$K : B ** i \bmod P$$

mientras para U2 la llave es:

$$K = A ** i' \bmod P$$

Un intruso poseyendo e, A, B y P no puede calcular prácticamente i o i' y, en consecuencia, no puede calcular K, y entonces esta llave de conexión se utiliza como un identificador mutuo. Si ahora U1 y U2 quieren generar una clave de comunicación particular para cada reinicialización de su comunicación, U1 genera al azar x, número impar de 210 bits y calcula K ** x mod P que transmite a U2 que procede de la misma forma. Entonces la llave de comunicación es para U1:

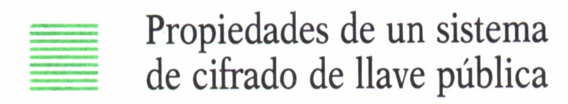
$$(K ** y) ** x \bmod P$$

y para U2:

$$(K ** x) ** y \bmod P$$

A continuación esta llave se utiliza para cifrar los datos transmitidos sobre la línea así reinicializada. Este cifrado puede hacerse con la ayuda de D.E.S., pues si el circuito se reinicializa cada minuto, el ataque brutal frente a D.E.S. es ineficaz.

Existen nuevos sistemas de clave pública en lo que puede probarse que su poder no depende de nada más que de la división en factores. Aunque parezca que dividir no es difícil, en el presente somos capaces de dividir números de 75 dígitos en un día, utilizando los más potentes computadores, pero la inclusión de 3 dígitos adicionales ya duplica el tiempo. Hace cinco años el número superior de dígitos que se podían dividir en un día era de 50, esos 25 dígitos de más que ahora podemos dividir se han conseguido, 15 de ellos, de avances matemáticos y 10, de la superior velocidad de los nuevos computadores. Indudablemente, con el avance de los métodos y la técnica, las posibilidades irán aumentando, lo que nos obliga a usar más dígitos en nuestros sistemas de cifrado y, por tanto, encarece el cómputo.



Propiedades de un sistema de cifrado de llave pública

Las propiedades de un sistema de cifrado de llave pública, definidas por Diffie y Hellman en el trabajo ya citado, son las siguientes:

Sea K un espacio finito de llaves, M un espacio finito de mensajes y (PK,SK) pares de transformaciones definidas en M. Entonces:

a) Para todas llaves, K, PK y SK son inversas.
b) Para toda llave K, todo mensaje m, PK(m) y SK(m) son fáciles de calcular.
c) Para casi todas las llaves K, es prácticamente imposible definir un algoritmo eficaz equivalente a SK conociendo solamente PK.
d) Para toda llave K, es fácil generar un par (PK,SK).

De este modo, cuando un usuario desea ser miembro de un sistema de cifrado de llave pública, utiliza un generador público de cifra de llave pública, para obtener un par (S,P). Guarda en secreto la transformación S, mientras que inscribe en el «armario» público la transformación P. Toda persona conociendo la transformación pública P puede dirigir un mensaje confidencial a ese usuario transmitiéndole el criptograma C = P(M). Unicamente quien conozca la transformación secreta S puede interpretar correctamente ese criptograma.

Un sistema de cifrado de llave pública permite igualmente obtener firmas, «matasellos», acuses de recibo, etc. Sólo quien posea la transformación secreta S puede emitir una firma R = S(M), pero todo el mundo puede verificar su autenticidad aplicando la transformación pública P.

Los dos principales generadores de llave pública, el de Rivest, el de Merkle y Hellman, se consideran a continuación.



Algoritmo RSA

Su principio consiste en utilizar un resultado clásico de la teoría de números, conocido con el nombre de teorema de Euler, el cual afirma que para evaluar la función de Euler de un número es necesario conocer su descomposición en factores primos. Ahora bien, es fácil mostrar que un número entero no es primo gracias al teorema de Fermat; sin embargo, es prácticamente seguro de que son primos y, por tanto, crear un gran número compuesto conteniendo dos grandes factores casi con certeza primos. Pero es harto imposible factorizar el gran número así obtenido sin idea previa de los factores que lo componen. Este algoritmo es utilizado por el Departamento de Defensa de los EE.UU. en aplicaciones de alta seguridad.

El algoritmo RSA, así llamado debido a las iniciales de sus autores (Rivest, Shamir y Aldeman), procede como sigue:

1. Para codificar el mensaje se utiliza una llave pública, formada por dos números enteros e y n.

2. Para decodificar el mensaje cifrado se utiliza una llave de descifrado privada, formada por dos números enteros d y n.

3. El mensaje se divide en n bloques de x dígitos, de forma que

$$M = d_1 d_2 ... d_x n.$$

4. El cifrado C del mensaje M será el resto de dividir M^e entre n. Es decir:

$$M^e \equiv r(\text{mód. } n)$$

Teniendo en cuenta que dos números enteros a y b son congruentes módulo m; es decir:

$$a = b(\text{mód } m)$$

Si se cumple que: a - b = xm, para algún x, entonces:

$$M^e \equiv r(\text{mód } n)$$

y denotando por E(M) el cifrado de M, será

$$C = E(M) = r$$

5. El descifrado de C, D(C) será el resto de dividir C^d entre n, por lo que:

$$C^d = b'n + r'$$
$$C^d = r'(\text{mód } n)$$
$$D(C) = r'$$

Se elige n como el producto de dos números primos p y q. Se recomienda escoger p y q de unos 100 dígitos, por lo que n será de unos 200 dígitos.

Se cumple que D (E(M)) = M

En efecto, sea (n) la función «cociente» de Euler para el número n.

Entonces,

$$\phi(n) = N(1 - 1/p)\ (1 - 1/q) = (p - 1)\ (q - 1)$$

Se escogen e y d tales que:

$$ed = 1\ (\text{mód. } \phi(n)).$$

Es decir,

$$ed = t\,\phi(n) + 1$$

Por otro lado,

$$M^{t\phi(n)+1} = M(\text{mód. } n)$$

En efecto:

a) Si M es relativamente primo a n

$$M^{t\phi(n)} = 1 \text{ (mód. n) y, por tanto,}$$
$$M^{t\phi(n)+1} \equiv M \text{ (mód. n)}$$

b) Si M no es relativamente primo a n, tendrá un factor de n, sea este p, entonces M = hp.

En este caso, q no puede ser factor de h, ya que M < n. Por tanto:

$$M \equiv 1 \text{(mód. q)}$$

Y también

$$M^{\phi(q)} = 1 \text{(mód. q)}$$

Siendo $\phi(q) = q = (1\text{-}1/q) = q\text{-}1$

Por las propiedades de las congruencias

$$M^{\phi(q)\phi(p)t} = \text{(mód. q)} \quad \text{o bien}$$
$$M^{\phi(n)t} = 1 + sq$$

y multiplicando por M se tiene que

$$M^{\phi(n)t+1} = M + sqhp = M + sh \quad \text{o bien}$$
$$M^{\phi(n)t+1} = M \text{ (mód. n)}$$

Análogamente, si se hubiera escogido q como factor de M, se habría llegado a

$$M^{t\phi(n)+1} = M \text{(mód. n)}$$

Así, pues,

$$M^{ed} \equiv M \text{ (mód. n)}$$

y también

$$M^{ed} = M + zn$$

Como

$$C^{d} = (M^{e}\text{-}bn)^{d} = b'n + r',$$

resulta que

$$M^{ed} = b''n + r'.$$

De donde

$$M = r'$$

Cumpliéndose, por tanto, que D (E(M)M) = M
Igualmente se cumple que E (D(M)) = M

Así, pues, el descifrado de C devuelve M. Por otro lado, a partir de la llave pública e, n no es posible obtener la llave privada d, n, ya que d no es fácilmente computable a partir de e y n.

La seguridad del esquema depende de la dificultad de factorizar un número n en producto de sus factores primos. El algoritmo de factorización más rápido que se conoce es el de Schroeppel, y da esta factorización en un número de pasos del orden:

$$\exp\ (\ln(n)\ln(\ln(n)^{0,5})$$

La complejidad de este problema de factorización, dependiendo del tamaño de los números p y q, es la siguiente:

Número de dígitos decimales	Número de operaciones	Tiempo
50	1,4 E 10	4 horas
75	9 E 12	104 días
100	2,5 E 15	74 años
200	1,3 E 23	4 billones de años

Actualmente este algoritmo está implementado sobre micros, donde parece que no es causa de graves problemas de uso.



Método de Merkle y Hellman

El principio del método se basa en el conocido problema de la «mochila», en inglés «knapsack», que puede enunciarse como sigue: siendo dada una longitud 1, y un conjunto finito de elementos de longitud l1, l2, ..., ln, encontrar un subconjunto de estos elementos que, puestos uno detrás de otro, dan exactamente la longitud 1.

Un ejemplo numérico podría ser: ¿se puede escribir 40 como la suma de ciertos números del conjunto {5,8,10,17,20,31}.

Obviamente, sí; 5+8+10+17 = 40. Así, 40 es la suma del primero, segundo, tercero y cuarto elemento del conjunto.

El algoritmo propuesto es como sigue:

a) Elegir dos números «suficientemente largos», m y n, de forma que exista un número p, con el cual: np ≡ 1 mód. m.

b) Se eligen l1, l2, ..., l100 y se transforman multiplicándolos por n y reduciéndolos módulo m.

c) Los números resultantes ki = nli mód. m, constituyen un conjunto aleatorio de números comprendidos entre 0 y m-1 y forman la base de un problema de «mochila» para quien no conozca los valores de m, n y p.

Este algoritmo es útil en un sistema de comunicaciones donde se guarde en secreto el conjunto de valores li, y los números m, n y p, mientras que el conjunto de ki se revela públicamente.

Los algoritmos de cifrado y descifrado son:

1. Hacemos público el conjunto k1, k2, ..., k100.
2. Si el mensaje a cifrar es t = (t1, t2, ..., t100).
3. El mensaje cifrado es MC = Eki(t) =

$$t_1k_1+t_2k_2+\ldots+t_{100}k_{100}$$

4. El descifrado de MC es D_{ki}(MC) = Texto claro.

Este algoritmo es más rápido que el RSA; en éste se necesitan 1.000 multiplicaciones en aritmética modular para cada operación y en el método de Merkle y Hellman sólo son necesarias 200 sumas para obtener el mismo fin.

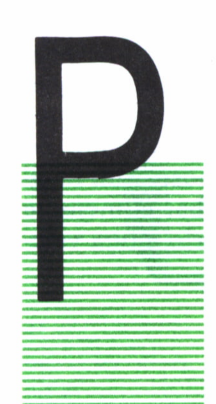


Problemas de los sistemas de llave pública

A pesar de las bondades pregonadas por sus inspiradores, los sistemas de llave pública presentan algunos inconvenientes que no es posible ignorar. En primer lugar, está el hecho de que estos sistemas son muy lentos de operación.

Por su parte, Kohnfelder apunta algún otro problema concerniente al uso de estos sistemas, como es la necesidad de una tercera parte fiable para operandos, so pena de que un intruso pudiera imitar la función del fichero público. También apunta que los sistemas de llave pública son más apropiados para transmisión de datos que para su almacenamiento.

NUEVAS TECNICAS DE SEGURIDAD 5



ARADOJICAMENTE hay que comenzar este capítulo comentando que los países denominados del tercer mundo escapan al espionaje electrónico de los países más avanzados, prescindiendo de toda tecnología; así, Irán entrega en mano los mensajes militares delicados.

Parece que hoy en día estos países están obligados a usar para sus mensajes el primer método de comunicación que se usó, y también el más barato y simple; debe ser más difícil introducirse en esos países y robar un mensaje a un emisario que captar vía satélite un mensaje y descifrarlo.

Pero, en general, no es así; la verdad es que cada día nacen nuevas técnicas y más sofisticadas para encubrir la información y más aún para impedir el acceso a los computadores y a las líneas de transmisión.

Indudablemente hoy día la información susceptible de ser cifrada reside normalmente en computadores, y el acceso a esta información puede hacerse o por uno de los terminales conectados a ese computador o cuando se transmite por una red de comunicaciones, pero también se encuentra en los microcomputadores de pequeñas empresas e incluso en los de profesionales, que lo usan como una herramienta de su trabajo.

Los computadores, normalmente, si lo hacen, solicitan una palabra de acceso, que es la clave que hay que introducir en el sistema, pero esa palabra normalmente no tiene más de ocho caracteres e incluso, en muchos casos, esta clave está definida como alfabética, por lo que el número de posibles claves es lo suficientemente restringido como para que cualquiera que se moleste un poco pueda acceder al sistema. Por ello, se está desarrollando una nueva técnica de criptografía denominada «Tecnología reloj»; en esta tecnología se hace uso de unos pequeños dispositivos independientes que no requieren modificación hardware del equipo y hacen las veces de cerradura y proporcionan una palabra de acceso diferente

cada vez que se usan; vamos a comentar a continuación algunos métodos de esta nueva tecnología, y algunas otras.



One-Time Pad

Este método es como sigue: se escribe el mensaje de forma clara y lo más breve posible; después, dicho mensaje se pone en una clave establecida entre emisario y receptor. A continuación es transmitido en Morse a un magnetófono, donde se graba en la cinta a una velocidad muy elevada. De este modo los puntos y rayas que constituyen el mensaje cifrado quedan comprimidos y se confunden en un «chirrido», que dura sólo unos segundos, que es el tiempo de transmisión a esta velocidad. El receptor no sólo debe disponer de la clave de cifrado, sino que deberá poseer un magnetófono de idénticas características al del emisor para grabar el mensaje a la velocidad en que le fue enviado y posteriormente pasarlo a una velocidad donde lo pueda comprender.

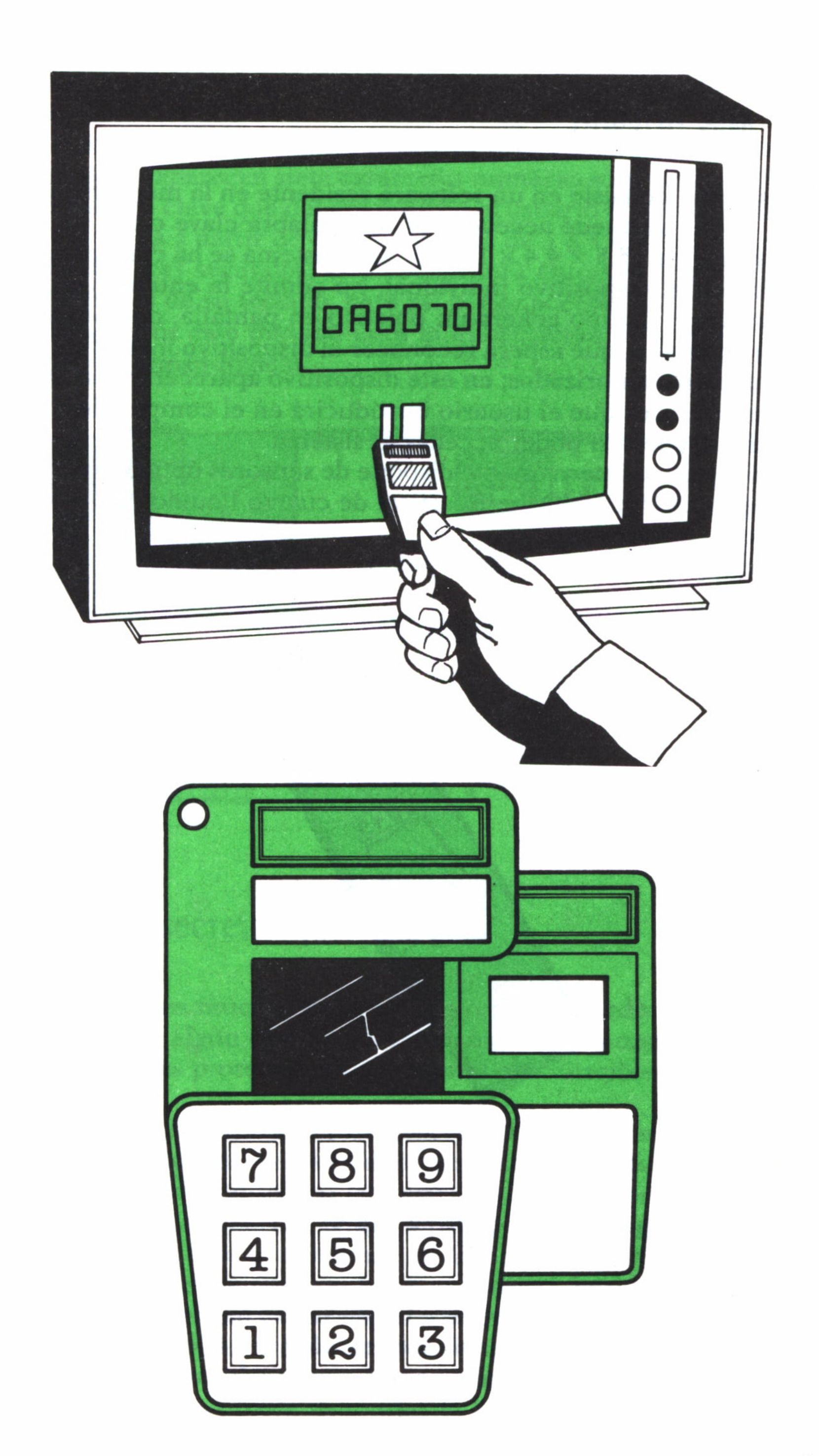


Llave de Acceso

Este método consiste en un dispositivo físico de 6,2 × 3,6 × 0,9 cm que trabaja conjuntamente con rutinas software residentes en el computador.

El dispositivo se sitúa delante del monitor al que se intenta acceder, recibe en un díodo que lleva incorporado las radiaciones que emite el monitor; esto produce una palabra de acceso que se visualiza en el dispositivo; el usuario teclea esa clave en el teclado del sistema, con lo que consigue el acceso; esta clave generada sólo es buena para una conexión, pues cada vez que se intenta el acceso es generada una clave distinta. Si este dispositivo le es sustraído a un usuario, lo comunica al gobernador del sistema, para que éste suprima de la memoria de acceso la clave del dispositivo y así quede inutilizable.

Este dispositivo internamente dispone de sensores ópticos, un circuito integrado CMOS/VLSI, una batería de litio, que dura una media de cinco años, un reloj (que es usado para modificar periódicamente el algoritmo de cifrado), contadores y un cristal de cuarzo líquido, donde se visualizan seis caracteres alfanuméricos. Este cristal muestra una clave generada por un dispositivo de 8.9 * 5.5 * 2.0 cm, donde se introduce el anterior cuando se intenta acceder a un dispositivo que no tenga pantalla; este otro dispositivo es una especie de calculadora con nueve teclas, que genera una clave a partir de un código numérico de identificación, visualizándose en la pantalla del dispositivo llave de acceso.


7 8 9
4 5 6
1 2 3

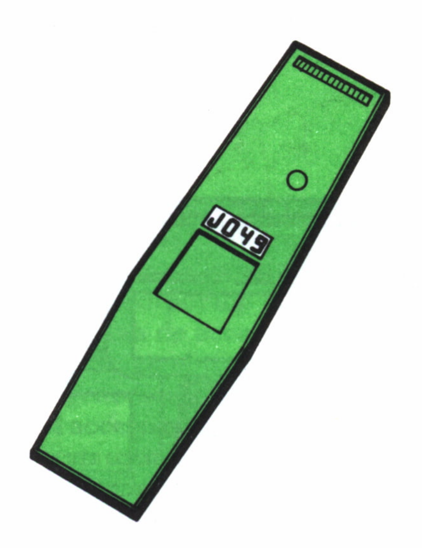


Más privado

Este sistema consiste en un software residente en la memoria del computador, al que se puede acceder por una palabra clave o por un dispositivo individual de 15,5 × 4,4 × 1,8 cm; si el sistema se ha configurado para ser usado con el dispositivo individual, no admite la entrada de ninguna palabra clave, el equipo genera un objetivo en pantalla, que no es visible para el ojo humano, que espera reconocer el dispositivo individual que poseen los usuarios autorizados; en este dispositivo aparecerán cuatro caracteres alfanuméricos que el usuario introducirá en el computador, a través de su teclado, para así poder acceder al sistema.

Este dispositivo internamente dispone de sensores ópticos, una batería, un microprocesador CMOS y un cristal de cuarzo líquido donde se visualiza el código a introducir.

Cada dispositivo construido es único.





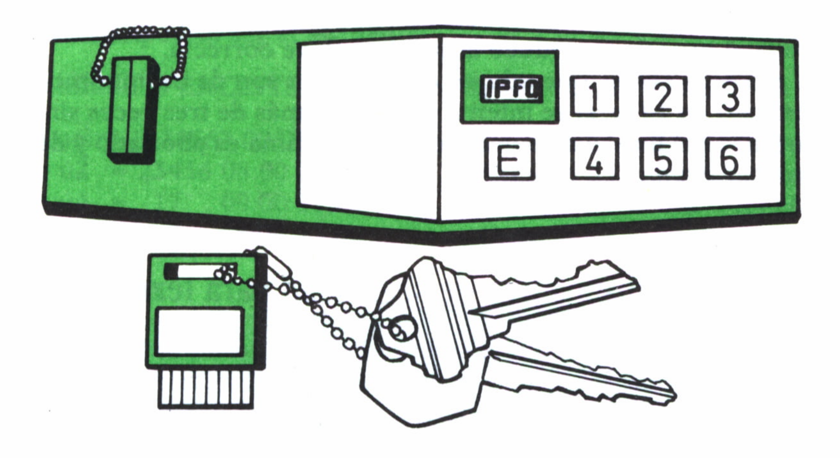
Sistema palabra segura DAS

Este sistema combina software y hardware; el hardware consiste en un dispositivo, parecido a una calculadora de 15,0 × 4,4 × 1,5 cm, y una llave de 2,0 × 2,0 × 0,9 cm, que son proporcionadas a los usuarios autorizados a acceder al sistema.

Este sistema opera en dos modos diferentes, dependiendo de cómo se configure.

En el primer modo, el usuario deberá insertar su palabra clave; en el segundo modo, aparecen siete caracteres numéricos en la pantalla, que indican la palabra clave que deberá insertar el usuario en su dispositivo DAS, el cual calculará a partir de esta clave otra, de 3 ó 4 dígitos, en un tiempo no superior a tres segundos; este código será entonces introducido en el sistema a través de su teclado, para así poder acceder a él.

El software de este método cifra la información a través de la aplicación de operaciones lógicas.





Disco secreto

Hoy día existen muchos programas para computadores que, vendidos en discos, llevan algún tipo de protección contra su copia, pero también es verdad que esas protecciones son fácilmente vencibles a poco que uno se lo proponga.

Este método, denominado Disco Secreto, supera de momento a los discos con protección que existen en el mercado, y lo consigue por su capacidad de crear discos «lógicos» (un disco lógico no existe en realidad, es, por decirlo así, imaginario), como una extensión del sistema operativo DOS. Por ejemplo, si se quiere crear un área o zona protegida en un disco; el sistema lo denomina como si fuera otro disco, esto hace que el acceso a este nuevo disco creado sea muy complicado.

Los datos que se almacenan en el disco protegido son automáticamente cifrados. Hay dos opciones para este cifrado; una, usando DES, y otra, usando un algoritmo propio denominado Método de Cifrado Rápido; esta

última opción consigue que el tiempo de almacenar un fichero en texto claro o cifrado sea prácticamente el mismo. En cambio, usando el DES, el tiempo se eleva notablemente.

Con este método el usuario puede crear:

a) Una o más áreas protegidas en un minidisco.
b) Un minidisco entero como un fichero protegido; y
c) Una o más áreas protegidas en un disco duro.

Cada área creada como área protegida, el sistema la toma como una unidad de disco diferente y con una palabra clave también distinta.

Como los datos según se van introduciendo se van criptografiando para su almacenaje, sólo se podrá acceder a ellos y visualizarlos en forma de texto claro si el usuario introduce la palabra clave correcta.

Las palabras claves que admite este sistema son de un mínimo de seis y un máximo de 24, y, si se intenta un acceso más de tres veces sin que se introduzca la palabra clave adecuada, el programa es abortado y devuelve el control al sistema.



TS 300 (dispositivo de seguridad para terminales)

Este dispositivo se conecta a la entrada de un terminal, entre éste y el computador central; los usuarios para acceder al sistema deben poseer una tarjeta, similar a las tarjetas de crédito, que es leída por el dispositivo a través de infrarrojos.

Estas tarjetas llevan, no visible al ojo humano, un código consistente en un número, que el dispositivo lee y envía al procesador central, el cual determina si está en su lista de usuarios autorizados y a qué información de la almacenada puede acceder; es posible añadir a la tarjeta cuatro dígitos más, que serán los que identifiquen al usuario; cuando se retira la tarjeta del dispositivo, se corta la comunicación con el procesador central. Es posible complicar el acceso todavía más, requiriendo el uso de dos tarjetas en vez de una.

Control de acceso limitado a microcomputadores por palabra clave, tiempo, día y/o programa.

Los microcomputadores actuales cuentan en su interior con una serie de conexiones donde se pueden insertar muy diversos tipos de dispositivos, denominados «placas», que realizan múltiples funciones; una placa de este tipo es la denominada ENIX.SYS, que puede realizar las siguientes funciones:

1. Doble palabra clave para acceder al sistema.
2. Limitación del acceso a días concretos de la semana y período de tiempo específico para usuarios individuales.

3. Cierre de seguridad para todos los elementos de un sistema (pantalla, teclado, etc.) cuando éste se deja desatendido.

4. Control de acceso al subdirectorio de programas.

5. Mensaje de comprobación y autentificación durante la telecomunicación.

6. Cifrado-descifrado de datos usando DES.

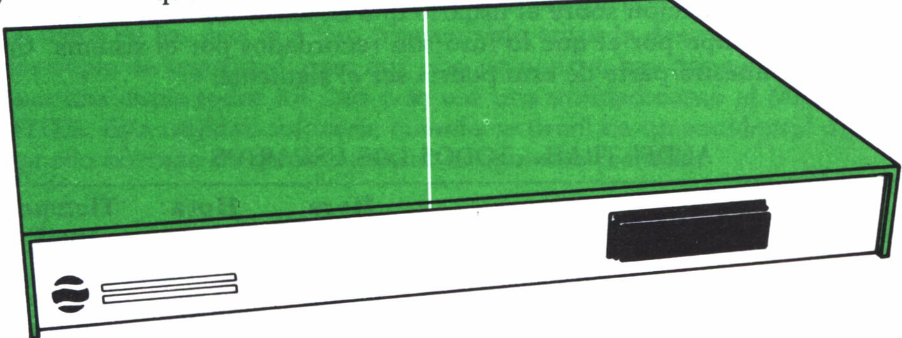
Este dispositivo hace que el usuario tenga que teclear una palabra clave y si ésta es correcta, tiene que teclear otra que, si es correcta, le permitirá el acceso al sistema; estas palabras claves pueden contener hasta 16 caracteres, incluyendo letras, números, signos especiales o de puntuación. Un listado de la zona oculta donde se reciben los datos de acceso y tiempos de trabajo podría ser el siguiente:

Bienvenido al ENIXSYS gobernador de la zona de tiempo

SysOp Login=	1234567	00:00	23:59	Usuario ú1 =	23456	08:00	17:30
Usuario ú2 =	23456	08:00	12:00	Usuario ú3 =	23456	12:00	17:00
Usuario ú4 =	17	09:00	17:00	Usuario ú5 =	6	09:00	14:00
Usuario ú6 =	7	08:00	12:00	Usuario ú7 =	23456	17:00	22:00



y nos indica que:



a) El gobernador (SysOp) del sistema tiene acceso, todos los días de la semana, 1234567, y todo el día, desde las 00:00 hasta las 23:59 horas.

b) El usuario 1 tiene acceso al microcomputador los días de la semana entre lunes y viernes, 23456, y de las 8:00 a las 15:30 horas, etc.

Este sistema posibilita el uso del microcomputador de 10 usuarios diferentes, pero puede ser ampliado si así se le solicita a la compañía que lo fabrica.

No sólo se restringe a los usuarios los días y tiempo de acceso, sino que también se les restringen los subdirectorios a que pueden acceder, y con dos posibles variantes:

a) Sólo leer.
b) Uso ilimitado.

El gobernador del sistema puede visualizar en un fichero especial hasta 99 intentos de acceso ilegales.



Detector de Huella

Este dispositivo es similar al anterior, en lo referente a que también es una placa que hay que conectar en el interior de un microcomputador. Esta placa, denominada PC/AUDIT, provee el acceso al sistema por nombre y palabra clave, así como el control de acceso a ficheros por software.

Este dispositivo contiene un reloj, memoria y una batería. En la memoria del dispositivo se almacena la lista de los usuarios autorizados a acceder al sistema y su palabra clave.

Al conectarse al sistema, éste solicita el nombre del usuario; si está en su memoria, solicita la palabra clave; si ésta es correcta, se le permite el acceso a los ficheros a que esté autorizado. Si en cinco minutos no se le proporciona al sistema ningún tipo de comunicación, la pantalla se queda en blanco, y sólo se volverá a activar si se vuelve a introducir la palabra clave.

Toda la información sobre el usuario que accedió, los ficheros en que trabajó y el tiempo por el que lo hizo son recordados por el sistema. Un listado que muestra parte de esto podría ser el siguiente:

AUDIT TRAIL - TODOS LOS USUARIOS

Usuario	**Clave**	**Fecha**	**Hora inicio**	**Hora fin**	**Tiempo total**
RAMON	AJ16‘	15/10/86	08:36a	10:20a	01:44
MANUEL	SOYYO	15/10/86	08:55a	12:00p	03:05
MARIA	TRASG	16/10/86	10:30a	11:45a	01:15

En él podemos ver el nombre del usuario, su clave, la fecha en que accedió, la hora de inicio del acceso, donde a significa la notación horaria A.M. y P.M., la hora de fin del acceso y el tiempo total de uso del sistema. La memoria de este dispositivo puede almacenar hasta 1.100 transacciones como las mostradas anteriormente; cuando la capacidad de esta memoria llega al 90%, un mensaje de aviso se visualiza en la pantalla para indicárselo al gobernador del sistema y que transfiera parte o la totalidad de los datos almacenados a otro dispositivo, para así poder seguir guardando los accesos que se le hagan; si no se hace así, cuando se llena empieza a escribir sobre la primera entrada que se introdujo.



Desayuno

Este método se usa para que cuando un usuario se ausenta de su puesto de trabajo para desayunar o por otro motivo cualquiera y está trabajando con datos confidenciales, no tenga que salirse de la aplicación específica que esté realizando , sino que con este método se bloquea el teclado y el sistema no puede ser usado hasta su vuelta, dejando en su pantalla un mensaje como máximo de 60 caracteres avisando de que no se puede utilizar el computador; cuando vuelve, teclea una clave de cinco dígitos y así se desbloquea el teclado y aparece la pantalla que tenía cuando se ausentó.

Recuperación de la información de un disco duro después de un atentado.

La información almacenada en un disco duro puede ser destruida accidentalmente o intencionadamente; en previsión de esto, las empresas suelen realizar una copia de esa información semanalmente a otro medio de archivo, por ejemplo, las cintas magnéticas; pero si la destrucción de la información tiene lugar el jueves de una semana y es el viernes el día que se realiza la copia de seguridad, esto obligaría a volver a introducir los datos de los días que han transcurrido entre la destrucción y la anterior copia de seguridad; para que esto no ocurra, hay dos soluciones, o se hace una copia todos los días o se usa una utilidad como el **MACE+UTILITIES**. Esta utilidad software, cuando se produce un accidental o intencionado borrado de datos, muestra la siguiente pantalla:

MACE+UTILITIES 32F/512K

F 1 - Ayuda	F 2 - Otras utilidades
F 3 - Diagnóstico	F 4 - Chkdsk
F 5 - Remedio	F 6 - Comprimir/clasificar directorios
F 7 - Condensar	F 8 - Crea archivo.M-U

Seleccione una opción

Si tecleamos F 8 visualizaremos:

F 1 - Ayuda	F 2 - Directorio
F 3 - Restaurar sistema	F 4 - Chkdsk
F 5 - Restaurar Archivo.M-U	F 6 - Reclamar
F 9 - Desborrar	F10 - Desformatear

donde vemos que tecleando F 5 podemos restaurar el archivo destruido.

Como hemos visto en estas posibles pantallas, esta herramienta tiene otras muchas posibilidades.

Todos estos métodos anteriores son susceptibles de tener fallos; unas veces, porque las claves pueden ser vistas u oídas por personas con acceso no autorizado, y otras, por la sustracción de los dispositivos individuales de acceso. Esto hará que el daño no sea detectado por el gobernador del sistema hasta que esté hecho, eso si se descubre.

Hoy en día hay otras técnicas más seguras que las anteriormente mencionadas, como son: reconocedores de huellas dactilares, geometría de la mano, forma de la palma de la mano, forma de los vasos sanguíneos de la retina, análisis de la voz, análisis de la firma.

Con estas técnicas se trata de hallar características individuales que no son iguales en dos individuos.

De estas técnicas, denominadas biométricas, quizá las dos más representativas y comerciales por su nivel de seguridad son:

a) Análisis de la huella dactilar.
b) Análisis de los vasos sanguíneos de la retina.

El problema de dispositivos que analicen estas características es su alto costo; ello impide que puedan ser usados en microcomputadores y queden reservados a grandes sistemas. Muchos dispositivos de este tipo tienen un valor superior a dos millones de pesetas.

Pero en grandes computadores, e incluso en el acceso a bases de datos confidenciales, este tipo de dispositivos pueden ser muy importantes.

Veamos más detalladamente algunos de estos métodos:



Detector de huella dactilar

El usuario debe teclear su número de identificación a través del teclado del sistema; si éste es correcto, se le indica que posicione el dedo sobre una lente. El sistema comprueba que esa huella dactilar corresponde al número de identificación introducido y si así es, se autoriza el acceso del usuario al sistema. El problema de estos dispositivos es su alto coste y el espacio de memoria que necesitan para almacenar los datos de cada usuario.



Geometría y tamaño de la mano

Este método es similar al anterior, con la diferencia de que se analizan o la forma de la mano o su tamaño, en vez de la huella dactilar. También

posee los mismos inconvenientes que el método anterior, respecto a necesidad de memoria y alto coste.



Reconocimiento de la voz

Este método todavía no es comercial; se basa en el estudio de las frecuencias que emite la voz del ser humano, que se comparan con las almacenadas en la memoria del sistema. Se desconfía de su precisión, y su coste será bastante elevado.



Análisis de la firma

Este método es el más antiguo de los comentados dentro de las técnicas biométricas, pues los primeros diseños se remontan a los años setenta.

El método se basa en que el usuario firma sobre una superficie destinada a tal fin, generalmente con un objeto especial; la firma pasa a ser analizada por el sistema, que decide si corresponde o no al usuario que ha solicitado el acceso.

Como las personas cambian la firma con el paso del tiempo, el sistema, si decide que una firma es correcta respecto al usuario que ha solicitado el acceso, la toma como modelo para el próximo acceso, con lo que actualiza su memoria constantemente; para el caso de las firmas ilegibles, que muchas veces construimos y el sistema no puede identificar, se le solicita al usuario una nueva firma; si después de un número de intentos, que se ha predeterminado, el sistema no reconoce la identificación de ese usuario, el sistema no admitirá a ese usuario el acceso hasta después de un tiempo también predeterminado.

Este método aporta unas ventajas interesantes, y son: su bajo coste, por una parte, y la comodidad para el usuario, por otra.

De este método hay varios enfoques para el análisis de la firma, entre los que hay que destacar:

— Posición del objeto de firma, mediante membranas de presión.
— Presión de la punta del objeto de firma, en las direcciones X, Y y Z.
— Aceleración en los trazos de la firma, mediante acelerómetros.

La captura de información en estos dispositivos se realiza por el objeto de la firma, la superficie de firma o ambas.



Identificador de la retina

Estos dispositivos piden la identificación de un usuario a través de su clave personal; posteriormente, si ésta se encuentra en la memoria del sistema, el usuario debe posicionar su ojo en un dispositivo que existe a tal fin; dicho dispositivo analiza si la retina visualizada corresponde a la clave introducida por el usuario. En caso de ser así, el acceso es permitido.

Algunos dispositivos de este tipo pueden almacenar hasta 1.200 análisis de usuarios autorizados; estos dispositivos pueden guardar en su memoria la retina de individuos que han intentado acceder sin estar autorizados.



BIBLIOGRAFIA

ALCOCER, M.: *Criptografía española*. Biblioteca Nacional. Madrid.

BENNETT, C. H.: «Quantum Cryptography and its Application to Probably Secure Key Expansion, Public-Key Distribution, and Coin-Yossing». *IEEE Abstract Int.* Symp. on Information Theory, pág. 91. Quebec. Septiembre, 1983.

BRILLOUIN, L.: *La Science et la Theorie de L'Information*. Mason. París, 1959.

BURTON, C. E.: «A Public Key Cryptography System». *Dr. Dobb's J.* Vol. 9, n.º 3, págs. 16-22. USA. Marzo 1984.

DESMEDT, Y. y cols.: "Cryptography Protects Information Agains Several Frands». Proc. of the Int. Carnaban Conf. on Security Techonology, págs. 255-259. Zurich. Octubre 1983.

DIFFIE, W., and HELLMAN, M. E.: «New Directions in Cryptography». *IEEE Trans. Inf. TH,* Vol. IT 22, n.º 6, págs. 644-654. Noviembre 1976.

EDP ANALYZER: «Data Encryption: Is it for You?» *Vista California*. Vol. 16, n.º 12. Diciembre 1978.

FEISTEL, H.: «Cryptography and Computer Privacy». *Scientific American,* Vol. 228, n.º 5, págs. 15-23. Mayo 1973.

FISHER, W. W.: «Cryptography for Computer Security: Making the Decision». *Comput. & Secur.* Vol. 3, n.º 3, págs. 229-233. Netherlands. Agosto 1984.

FORSYTH, F.: *The Fourth Protocol*. Hutchinson. Londres 1984.

GOVAERTS, R. J. M. y cols.: «Cryptography: How to Attack, What to Protect?» Proc. of the International Conf. on Comm. ICC84, Vol. 1, págs. 175-178. North-Holland. Amsterdam. Mayo 1984.

GREENBERGER, M.: *JACM 8*. págs. 383-389. 1961.

GREENWOOD, G.: *The Micro Cloak and Dagger Book*. Sigma Press. 1984.

GUILLON, L. C. and LORING, B.: «The Impact of Cryptography in the Desing of the New Services». *Proc. ICCC* 78, págs. 303-308. Kyoto, 1978.

HIGHLAND, H. J.: «Random Bits and Bytes». *Compt. & Secur.* Vol. 5, págs. 3-9. North-Holland. Amsterdam, 1986.

HIGHLAND, H. J.: «Random Bits and Bytes». *Compr. & Secur.* Vol. 5, págs. 85-100. North-Holland. Amsterdam, 1986.

HIGHLAND, H. J.: «Random Bits and Bytes». *Compr. & Secur.* Vol. 5, págs. 181-192. North-Holland. Amsterdam, 1986.

KAFKA, G.: «Data Encryption: Cryptography Offers Effective Protection Against Data Theft». *Elektronik*. Vol. 26, n.º 12, págs. 159-163. Junio 1984.

KHAN, D.: *The Codebreakers*. McMillan. New York, 1973.

LENNON, R. E. y cols.: «Crytography in Data Processing». *Data Procesing*. Vol. 9, n.º 7, págs. 36-38. GB. Septiembre 1984.

LOBEL, J.: *Foiling the Systems Breakers*. McGraw-Hill. New York, 1986.

LOGPRE, L.: «The Use of Public-Key Cryptography System». *Dr. Dobbs'J*. Vol. 9, n.º 3, págs. 16-22. USA. Marzo 1984.

MERKLE, R. C. y HELLMAN, M. E.: «Hiding Information and Signatures in Trapdoor Knapsacks». *IEEE Trans. Inf. Th*, Vol. IT24, n.º 5, págs. 525-530. Septiembre 1978.

MEYER, C. H.: «Desing Considerations for Cryptography». *AFIPS. Con. Proc.* 42, págs. 603-606. 1973.

MEYER, C. H. y cols.: *Cryptography: A New Dimension in Computer Data Security*. John Wiley and Sons. New York, 1982.

NBS: «Data Encryptino Standar». *FIPS Pub*. 46. Enero 1977.

POHLING, S., y HELLMAN, M. E.: «An Improved Algorithm for Computig Logarithm Over GF (P)». *IEEE Trans. Inf. Th*. Vol. IT24, n.º 1, págs. 106-110. Enero 1977.

RANDELL, S.: «The Colossus». Proc. Int. Conf. an History of Computing. Los Alamos, 1976.

RIVEST, R. L. y cols.: «A Method for Obtaining Digital Signatures and Public Key Cryptosystems». Comm. *ACM*. Vol. 21, n.º 2, págs. 120-126. Febrero 1987.

ROBLING, E. D.: «Cryptography and Data Security». Addison Wesley. *P. C. Reading*. Massachusets, 1982.

RODRÍGUEZ, A.: *Protección de la Información*. Paraninfo. Madrid, 1986.

SANCHO, J.: *Diseño de métodos criptográficos para la protección de la información en sistemas ordenadores*. Tesis Doctoral. Madrid, 1979.

SHANNON, C. E.: «Comunication Theory of Secrety Systems». *BSTJ*. Vol. 28, págs. 665-775. 1949.

SUGARMAN, R. M.: «Freedom to Research and Publish on Cryptography Remains Unresolved». *The Institute News-Supp to Spectrum*. Vol. 2, n.º 5. 1978.

TUCHMAN, W.: «Proc. National Computer Conference». AFIPS 1978. VANDEWALE, J., y GOVAERTS, R.: «Does Public-Key Cryptography Provide a Practical and Secure Protection of Data Storage and Transmission?». Proc. of the Int. Carnahan Conf. on Security Technology, págs. 113-119. Zurich. Agosto 1983.

WIDMAN, K. O.: «Cryptography. The Art of Secret-Keeping». Proc. Int. Zurich Seminar on Digital Comm, págs. 151-156, Zurich. Marzo 1984.

ENCICLOPEDIA PRACTICA DE LA

INFORMATICA APLICADA

INDICE GENERAL

5 AÑADA PERIFERICOS A SU ORDENADOR

Breve descripción de varios periféricos que facilitan la comunicación con el ordenador personal, con algunos ejemplos de fácil construcción: ratón, lápiz óptico, marco para pantalla táctil...

6 GRAFICOS ANIMADOS CON EL ORDENADOR

En este libro las técnicas utilizadas para la animación son el resultado de unas pocas ideas básicas muy sencillas de comprender. Descubrirá los trucos y secretos de movimientos, choques, rebotes, explosiones, disparos, saltos, etc.

7 JUEGOS INTELIGENTES EN MICROORDENADORES

Los ordenadores pueden enfrentarse de forma «inteligente» ante puzzles y otros tipos de juegos. Esto es posible gracias al nuevo enfoque que ha dado la IA a la tradicional teoría de juegos.

8 PERIFERICOS INTERACTIVOS PARA SU ORDENADOR

Descripción detallada de la forma de construir, paso a paso y en su propia casa, dispositivos electrónicos que aumentarán la potencia y facilidad de uso de su ordenador: tableta digitalizadora, convertidores de señales analógicas, comunicaciones entre ordenadores.

9 COMO HACER DIBUJOS TRIDIMENSIONALES EN EL ORDENADOR

Compruebe que también con su ordenador personal puede llegar a diseñar y calcular imágenes en tres dimensiones con técnicas semejantes a las utilizadas por los profesionales del dibujo con equipos mucho más sofisticados.

10 PRACTIQUE MATEMATICAS Y ESTADISTICA CON EL ORDENADOR

En este libro se repasan los principales conceptos de las Matemáticas y la Estadística, desde un punto de vista eminentemente práctico y para su aplicación al ordenador personal. Se basan los diferentes textos en la presentación de pequeños programas (que usted podrá introducir en su ordenador personal).

11 CRIPTOGRAFIA: LA OCULTACION DE MENSAJES Y EL ORDENADOR

En este libro se presentan las técnicas de ocultacion de mensajes a través de la criptografía desde los primeros tiempos hasta la actualidad, en que el uso de los computadores ha proporcionado la herramienta necesaria para llegar al desarrollo de esta ciencia.

12 APL: LENGUAJE PARA PROGRAMADORES DIFERENTES

APL es un lenguaje muy potente que proporciona gran simplicidad en el desarrollo de programas y al mismo tiempo permite programar sin necesidad de conocer todos los elementos del lenguaje. Por ello es ideal para quienes reúnan imaginación y escasa formación en Informática.

13 PRACTIQUE CIENCIAS NATURALES CON EL ORDENADOR

Ejemplos sencillos para practicar con el ordenador. Casos curiosos de la Naturaleza en forma de programas para su ordenador personal.

14 COMO SIMULAR CIRCUITOS ELECTRONICOS EN EL ORDENADOR

Introducción a los diferentes métodos que se pueden emplear para simular y analizar circuitos electrónicos, mediante la utilización de diferentes lenguajes.

15 LOS LENGUAJES DE LA INTELIGENCIA ARTIFICIAL

Libro en que se describen los lenguajes específicos para la «elaboración del saber» y los entornos de programación correspondientes. El conocimiento de estos lenguajes, además de interesante en sí mismo, es sumamente útil para entender todo lo que la Informática Artificial supondrá para el futuro de la Informática.

16 PRACTIQUE FISICA Y QUIMICA CON SU ORDENADOR

Libro eminentemente práctico para realizar pequeños «experimentos» con su ordenador y distraerse de un modo útil.

17 EL ORDENADOR Y LA LITERATURA

En este libro se examinan procesadores de textos, programas de análisis literario y una curiosa aplicación desarrollada por el autor: APOLO, un programa que compone estructuras poéticas.

18 COMO ELEGIR UNA HOJA ELECTRONICA DE CALCULO

En este título se estudian las diferentes versiones existentes de esta aplicación típica, desde el punto de vista de su utilidad para, en función de las necesidades de cada usuario y del ordenador de que dispone, poder elegir aquella que más se adecúe a cada caso.

19 ECONOMIA DOMESTICA CON EL ORDENADOR PERSONAL

Breve introducción a la contabilidad de doble partida y su aplicación al hogar, con explicaciones de cómo utilizar el ordenador personal para facilitar los cálculos, mediante un programa especialmente diseñado para ello.

20 ¿MAQUINAS MAS EXPERTAS QUE LOS HOMBRES?

Después de situar los «sistemas expertos» en el contexto de la inteligencia artificial y describir su construcción, su funcionamiento, su utilidad, etc., se analiza el papel que pueden tener en el futuro (y presente, ya) de la Informática.

21 PRACTIQUE HISTORIA Y GEOGRAFIA CON SU ORDENADOR

Libro interesante para los aficionados a estas ciencias, a quienes presenta una nueva visión de cómo utilizar el microordenador en su estudio.

22 ERGONOMIA: COMUNICACION EFICIENTE HOMBRE-MAQUINA

Análisis de la comunicación entre el hombre y la máquina, y estudio de diferentes soluciones que tienden a facilitarla lo más posible.

23 EL ORDENADOR Y LA ASTRONOMIA

Los cálculos astronómicos y el conocimiento del firmamento en un libro apasionante y curioso.

24 VISION ARTIFICIAL. TRATAMIENTO DE IMAGENES POR ORDENADOR

El procesado de imágenes es un campo de reciente y rápido desarrollo con importantes aplicaciones en áreas tan diversas como la mejora de imágenes biomédicas, robóticas, teledetección y otras aplicaciones industriales y militares. Se presentan los principios básicos, los sistemas y las técnicas de procesado más usuales.

25 LA ESTACION TERMINAL PERSONAL

Las modernas técnicas de comunicación van permitiendo que las grandes capacidades de proceso y el acceso a bases de datos de gran tamaño estén cada día más al alcance de cada usuario (fuera ya de los Centros de Proceso de Datos).

26 EL ORDENADOR COMO MAQUINA DE ESCRIBIR INTELIGENTE

Descripción de los sistemas de tratamiento de textos existentes, análisis comparativos y estudio de posibilidades de cada uno de ellos. Guía práctica para la elección del presente paquete que más se adecúe a nuestras necesidades y al ordenador personal de que dispongamos.

27 EL LENGUAJE C, PROXIMO A LA MAQUINA

Lenguaje de programación que se está imponiendo en los microordenadores más grandes, tanto por su facilidad de aprendizaje y uso, como por su enorme potencia y su adecuación a la programación estructurada. Vinculado íntimamente al sistema operativo UNIX es uno de los lenguajes de más futuro entre los que utilizan los micros personales.

28 EL ORDENADOR COMO INSTRUMENTO MUSICAL Y DE COMPOSICION

Análisis de cómo se puede utilizar el ordenador para la composición o interpretación de música. Libro eminentemente práctico, con numerosos ejemplos (que usted podrá practicar en su ordenador casero) y lleno de sugerencias para disfrutar haciendo de su ordenador un verdadero instrumento musical.

29 LA CREATIVIDAD EN EL ORDENADOR. EXPERIENCIAS EN LOGO

El LOGO es un lenguaje enormemente capacitado para la creación principalmente gráfica y en especial para los niños. En este sentido se han desarrollado numerosas experiencias. En el libro se analizan estas experiencias y las posibilidades del LOGO en este sentido, así como su aplicación a su ordenador casero para que usted mismo (o con sus hijos) pueda repetirlas.

30 SISTEMAS OPERATIVOS: EL SISTEMA NERVIOSO DEL ORDENADOR

Características de diversos sistemas operativos utilizados en los ordenadores personales y caseros. Se trata de llegar al conocimiento, ameno, aunque riguroso, de la misión del sistema operativo de su ordenador, para que usted consiga sacar mayor rendimiento a su equipo.

NOTA: Ediciones Siglo Cultural, S. A., se reserva el derecho de modificar, sin previo aviso, el orden, título o contenido de cualquier volumen de la colección.

En este libro se ha tratado de recoger, a un nivel comprensible para no avanzados en la materia, distintos sistemas de seguridad, desde los más remotos que se mencionan en la introducción, hasta las últimas novedades en sistemas de seguridad.

El libro incluye algunos programas de los métodos descritos. Estos programas han sido realizados en Basic de Microsoft y Turbo Pascal. Pueden ejecutarse sobre computadores que dispongan de los sistemas operativos MS-DOS y CP/M.

En el capítulo de Técnicas avanzadas se han descrito las fórmulas sobre las que se fundamentan los algoritmos que en él se describen; pese a que estas fórmulas son relativamente complejas, ha sido necesaria su inclusión para aumentar la comprensión sobre estos métodos, para aquellos que quieran profundizar en la materia.